

Trabajo de fin de Máster en Ingeniería Informática

Plataforma NIRS PAT para la industria 4.0 con Tensorflow

Autor : Víctor Chamizo Álvarez

Fecha de la defensa : 31 Enero 2019

Director: Alejandro A. Rosales Lavielle,
IRIS Technology Group

Ponente: Marta Arias Vicente,
Departamento: Ciencias de la Computación

Titulación: Máster en Ingeniería Informática

Centro: FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Universidad: UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) BarcelonaTech

Resumen

El concepto Industria 4.0 busca conseguir una nueva manera de organizar los medios de producción, consiguiendo “fábricas inteligentes”. Una forma de llegar a ello es con las tecnologías analíticas de proceso, PAT por sus siglas en inglés, mediante la espectroscopía de infrarrojo cercano, y el análisis quimiométrico para obtener parámetros críticos que afectan a la calidad. Conseguirlo precisa de nuevos métodos analíticos más potentes, por lo que el objetivo de esta tesis ha sido usar la plataforma de Tensorflow, ampliamente usada en deep learning por su facilidad de gestión de GPUs para cálculo, para compararla con los métodos analíticos tradicionales en la quimiometría como PLS y PLS-DA. Los resultados obtenidos con las redes neuronales creadas con esta plataforma han conseguido varias mejoras entre ellas, la predicción con exactitud del 99.93% con los datos de test en clasificación a diferencia del 44% obtenido con PLS-DA. También una reducción del RMSE con los datos de test en cuantificación de un 1.91 obtenido con PLS a 0.96 con redes convolucionales. Y por último llegar a clasificar 3 tipos de cuerpos extraños en imagen hiperespectral cuando estos están a una profundidad de 4 mm, mientras que con PLS-DA no se consigue detectar ninguno.

Abstract

The Industry 4.0 concept seeks to achieve a new way of organizing the means of production, achieving "smart factories". One way to achieve this is with PAT, process analytical technologies, using near infrared spectroscopy, and chemometric analysis to obtain critical parameters that affect quality. Reach this requires the use of more powerful analytical methods, so the objective of this thesis has been to use the Tensorflow platform, widely used in deep learning for its ease of management of GPUs for calculation, to compare it with traditional analytical methods in chemometrics as PLS and PLS-DA. The results obtained with the neural networks created with this platform have achieved several improvements among them, the prediction with 99.93% accuracy with the test data in classification, unlike the 44% obtained with PLS-DA. Also a reduction of the RMSE with the test data in quantification from a 1.91 obtained with PLS to 0.96 with convolutional networks. And finally get to classify 3 types of foreign bodies in hyperspectral image when they are at a depth of 4 mm, while with PLS-DA it is not possible to detect any.

Agradecimientos

He de agradecer a mis compañeras de trabajo: Inmaculada Campos, a Idoia Martí i Aluja y a Kudirat Obisesan por los consejos que me han dado y sobretodo por proporcionarme los valiosos datos, muy tediosos de adquirir, que he usado en este trabajo. También a mi ponente, Marta Arias Vicente, por indicarme algunos gazapos técnicos al revisar el escrito, con los que he conseguido unos resultados más acordes a la realidad después de arreglarlos. Y por último y no menos importante, agradezco a mi director Alejandro A. Rosales por sus contribuciones y por el hecho de que me apoyara desde el principio en esta exploración de nuevos métodos analíticos.

Índice

Introducción	4
Objetivos	6
Estado del arte	7
Espectroscopía de un píxel	8
Descripción de los datos	8
Análisis de datos	9
Clasificación usando absorbancia	10
Preprocesado	12
Red neuronal densamente conectada	14
Red convolucional	15
Clasificación usando raw data	17
Preprocesado	17
Red neuronal densamente conectada	18
Red convolucional	19
Comparativa PLS-DA vs Redes neuronales artificiales	19
Cuantificación usando absorbancias	21
Red neuronal densamente conectada	22
Red convolucional	24
Cuantificación usando raw data	26
Red neuronal densamente conectada	26
Red convolucional	30
Comparativa PLS vs Redes neuronales artificiales	32
Imagen hiperespectral	34
Descripción de datos	34
Análisis de datos	35
Extracción datos para modelar	35
Clasificación usando absorbancia	38
Red neuronal densamente conectada	38
Red convolucional	41
Comparativa DNN vs CNN	43
Trabajo futuro	45
Conclusiones	46
Referencias	47
Glosario	49

Introducción

Una nueva manera de organizar los medios de producción es lo que se pretende con el concepto “Industria 4.0”. El objetivo que pretende alcanzarse es la puesta en marcha de un gran número de “fábricas inteligentes” capaces de una mayor adaptabilidad a las necesidades y a los procesos de producción, así como a una asignación más eficiente de los recursos, abriendo así la vía a una nueva revolución industrial[1].

Las tecnologías analíticas de proceso, en inglés PAT, han sido definidas por la Administración de Alimentos y Fármacos, FDA, de los Estados Unidos como mecanismos para diseñar, analizar y controlar procesos de fabricación farmacéuticos a través de la medición de parámetros críticos(CPP) que afectan a los atributos de calidad (CQA) [2]. Para conseguir implementar proyectos PAT de forma satisfactoria es esencial utilizar las siguientes herramientas:

- Adquisición de datos multivariable y herramientas de análisis de datos: usualmente paquetes de software avanzados que ayudan en el diseño de experimentos, la recopilación de datos sin procesar y el análisis estadístico de estos datos para determinar qué parámetros son CPP.
- Herramientas de química analítica de procesos: instrumentos analíticos en línea utilizados para medir los parámetros que se han definido como CPP. Estos incluyen principalmente espectroscopia de infrarrojo cercano (NIRS); pero también incluyen biosensores, espectroscopia Raman, fibra óptica y otros.
- Herramientas de mejora continua y / o gestión de conocimiento: sistemas en papel o paquetes de software que acumulan datos de Control de Calidad adquiridos a lo largo del tiempo para procesos específicos con el objetivo de definir las debilidades del proceso e implementar y monitorear iniciativas de mejora de procesos. Estos productos pueden ser iguales o separados de las herramientas de análisis estadístico anteriores.

En resumen las PAT vienen a ir enfocadas en una serie de mejoras a largo plazo a las que cualquier industria quiere llegar, tales como : reducir el tiempo de producción, prevenir el rechazo de lotes de producción, habilitar el lanzamiento de producto en tiempo real, mejorar la eficiencia energética y de materiales, y facilitar el procesado continuo.

El enfoque que tiene IRIS Technology solutions sobre qué es el paradigma Industria 4.0 en lo que a control de procesos industriales se refiere:

- Disponer de suficiente cantidad de datos informativos y fiables no sólo sobre las condiciones externas del proceso sino sobre todo de lo que ocurre dentro del proceso, o sea, una combinación de sensórica y PAT.
- Desvelar información relevante oculta en los datos mediante recursos TIC fundamentados en herramientas de inteligencia artificial a fin de construir un sistema de soporte de decisiones libre de sesgos cognitivos y prejuicios.

Una de las tecnologías esenciales para llegar a estos objetivos es la espectroscopia del infrarrojo cercano, NIRS por sus siglas en inglés, es una técnica robusta, precisa y no destructiva para monitorizar parámetros físicos y químicos en el procesado de productos alimenticios o farmacéuticos, que está bastante aceptada como la técnica más prometedora para el control de procesos on/in-line [3].

Puesto que el deep learning y en concreto el uso de las redes neuronales ha sido y sigue siendo a día de hoy, una tecnología disruptiva parece necesario integrarla en el ecosistema presentado. Con el fin de seguir avanzando en la nueva etapa industrial con sistemas más precisos y por lo tanto que aportan más seguridad a la producción alimentaria o farmacéutica y por ende al consumidor final.

Objetivos

Ver si nuevos métodos de machine learning, como el deep learning, sirven para modelizar NIRS y mejoran los métodos clásicos como PLS para cuantificación y PLS-DA para clasificación.

Usar una plataforma escalable a los nuevos paradigmas de medición en la espectroscopía de infrarrojo cercano como la imagen hiperespectral, dado el gran volumen de datos que genera se precisa que la infraestructura de análisis sea capaz de aprovechar la potencia de cálculo de las GPU.

La quimiometría requiere de una calibración para aplicar más eficientemente sus métodos analíticos. Esto es un impedimento a la hora de desarrollar sistemas de monitorización que funcionen sin pausa, por lo que habría que comprobar que las redes neuronales son capaces de predecir eficientemente con los datos raw.

Conseguir muchas muestras con los rangos de calidad a medir no siempre es posible, por lo que hay que comprobar si las redes neuronales son efectivas aun cuando no se tiene un gran volumen de datos.

En definitiva el objetivo final es conseguir un sistema de predicción más certero y escalable, el cual es clave en los requisitos que se proyectan en las PAT y para la industria 4.0. Tener un control del estado de un proceso industrial ya sea un mezclado, o la calidad de una materia prima, o el estado final de un producto es sumamente necesario, ya que ahorra costes y aumenta la seguridad hacia el consumidor final.

Estado del arte

En la actualidad los dispositivos de monitorización desarrollados en IRIS Technology Solutions S.L utilizan tecnología NIRS empleando la quimiometria como base para el análisis de los datos [4]. La quimiometría es la ciencia de extraer información de sistemas químicos por medio de datos. La quimiometría se aplica para resolver problemas descriptivos y predictivos en ciencias naturales experimentales, especialmente en química. En las aplicaciones descriptivas, las propiedades de los sistemas químicos se modelan con la intención de aprender las relaciones subyacentes y la estructura del sistema, es decir, la comprensión e identificación del modelo. En aplicaciones predictivas, las propiedades de los sistemas químicos se modelan con la intención de predecir nuevas propiedades o comportamientos de interés. En ambos casos, los conjuntos de datos pueden ser pequeños, pero a menudo son muy grandes y muy complejos, e involucran de cientos a miles de variables y de cientos a miles de casos u observaciones.

Las técnicas de clasificación multivariada supervisadas están estrechamente relacionadas con las técnicas de calibración multivariada en el sentido de que se utiliza un conjunto de entrenamiento o calibración para desarrollar un modelo matemático capaz de clasificar muestras futuras. Las técnicas empleadas en la quimiometría son similares a las utilizadas en otros campos: análisis discriminante multivariado, regresión logística, redes neuronales, árboles de regresión/clasificación. El uso de técnicas de reducción de rango junto con estos métodos de clasificación convencionales es rutinario en quimiometría, por ejemplo, análisis discriminante en componentes principales o puntajes de mínimos cuadrados parciales.

Los métodos utilizados con los datos que se presentan en este documentos han sido PLS-DA y PLS respectivamente, por lo que estos métodos, y más concretamente los modelos de predicción obtenidos, por el área de quimiometría de IRIS, son los que se comparan con las redes neuronales. La regresión de mínimos cuadrados parciales o Partial least squares regression, PLS por sus siglas en inglés, es un método estadístico que tiene relación con la regresión de componentes principales. En lugar de encontrar hiperplanos de máxima varianza entre la variable de respuesta y las variables independientes, se encuentra una regresión lineal mediante la proyección de las variables de predicción y las variables observables a un nuevo espacio [5]. Debido a que tanto los datos de X e Y se proyectan a nuevos espacios, la familia de los modelos PLS se conoce como factor de modelos bilineales. Los cuadrados mínimos parciales Análisis discriminante (PLS-DA) es una variante que se utiliza cuando la Y es categórica. El principal inconveniente de estas técnicas es que gestionan mal los datos no lineales, aunque se ha demostrado su capacidad para predecir en casos no lineales a base de usar más variables latentes con el riesgo añadido de modelar el ruido junto con la variabilidad correcta. Por esta razón buscar una alternativa más poderosa a estas técnicas es requerida.

En cuanto a las redes neuronales, se componen de conjuntos de neuronas, las cuales cada una contiene una función de activación. Las neuronas tienen conexiones de entrada y salida con sus respectivos weights y bias. Los weights muestran la fortaleza con la conexión, es decir, si se incrementa la entrada cuánto influye en la salidas. Por otro lado los bias muestran cómo de lejos están nuestras predicciones de los valores reales. Un espectro puede ser visto como una imagen unidimensional, y eso es un tipo de arquitectura de redes neuronales que se ha usado en datos espectroscópicos. Visto de esta forma ya se han implementado redes de clasificación para datos obtenidos en la espectroscopia Raman, consiguiendo clasificar tabletas farmacéuticas usando espectroscopia VIS-NIR [6,7].

Espectroscopía de un píxel

Descripción de los datos

Los datos tratados se componen en dos experimentos, uno son datos obtenidos por medir diferentes tipos de talcos y los otros son datos obtenidos de medir carne picada usada en la elaboración de embutido. Estos datos se capturaron usando analizadores portátiles NIRS, en este caso Visum Palm de 512 canales. Este dispositivo adquiere el espectro de la muestra y registra el dato raw y la absorbancia. Un dato raw se compone por 512 valores enteros de 0 a 65535, cada uno de estos valores corresponde a la suma total de fotones recibidos para cada una de las longitudes de onda discretizadas desde 900 nm a 1700 nm.



Figura 1: Analizador NIRS portátil Visum Palm

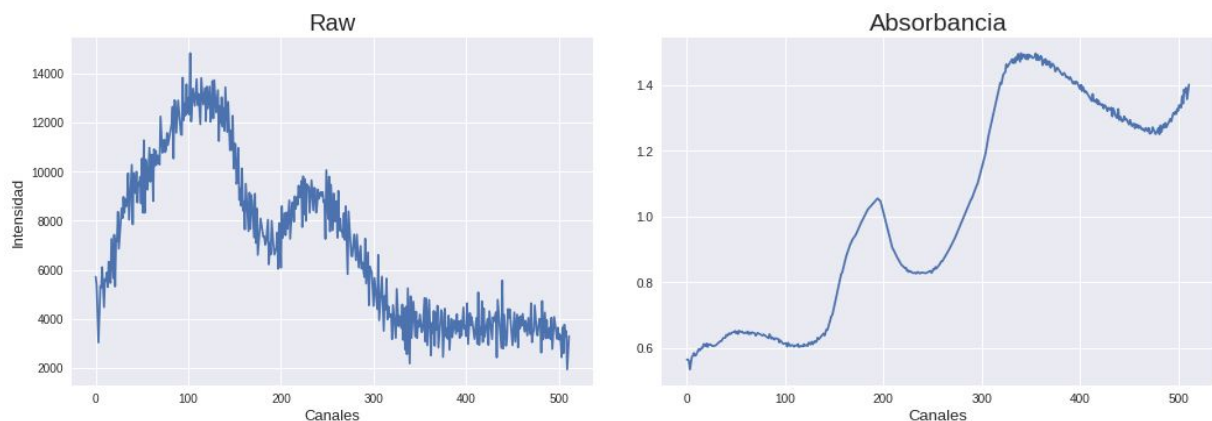


Figura 2: espectro raw a la izquierda y su transformada en absorbancia a la derecha

En espectrofotometría, la absorbancia se define como $A_{\lambda} = -\log_{10}(I/I_0)$ donde I es la intensidad de la luz con una longitud de onda específica tras haber atravesado una muestra, I_0 es intensidad de la luz antes de entrar a la muestra (intensidad de la luz incidente). Para su cálculo se precisa una calibración inicial del dispositivo donde se muestrea un material de alta reflectancia conocida. Con ella se obtiene un bright que viene a ser la cota superior de lo que se espera como máximo adquirir. Luego se toma una muestra sin

luz, llamada dark, que es la cota inferior, básicamente el ruido intrínseco del sensor. Aplicando la siguiente fórmula se obtiene la absorbancia de cada raw data.

$$absorbancia(i) = \log\left(\frac{bright(i)-dark(i)}{raw(i)-dark(i)}\right)$$

Análisis de datos

Los resultados de referencia a superar han sido obtenidos con los modelos quimiométricos usando las absorbancias. Para el caso de los talcos se requiere clasificación por lo que PLS-DA ha sido el método mientras que con la carne se usó PLS para cuantificar una serie de parámetros de la muestra como humedad, proteína, cantidad de sal y grasa.

Para entrenar las redes se han usado tanto las absorbancias como los datos raw del del sensor. Los datos son leídos de un archivo csv, del inglés valores separados por comas, donde cada fila representa un espectro y su clase o valor asociado. Tanto para generar el modelo de clasificación como los de cuantificación se ha optado por separar el conjunto de datos en train y test con una proporción $\frac{2}{3}$ y $\frac{1}{3}$ respectivamente. Se han usado utilidades de Python de la librería Scikit Learn[8] como el *StratifiedShuffleSplit*, un objeto para validación cruzada, con el cual se obtiene subconjuntos de datos aleatorios pero que conservan el porcentaje para cada clase. Aunque no estamos manejando conjunto de datos muy elevados se ha optado por no recurrir a métodos de validación cruzada como K-fold Cross-Validation o Leave-One-Out Cross-Validation puesto que entrenar una vez el modelo requiere de mucho tiempo. La validación cruzada o cross-validation es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones.

Todos los resultados obtenidos han sido usando el framework the Google Collaboratoy, una PaaS online que integra el backend the Tensorflow[9] v1.12.3 en un sistema de ejecución de cuadernos al estilo Jupyter. El lenguaje de programación usado ha sido Python 3.6 y para la creación de las redes se ha usado Keras[10,11] puesto que ofrece compatibilidad con varios backends como Caffe y Theano. Otro punto interesante a destacar es que Collaboratory permite que entrenes las redes con la GPU local de tu máquina o usando GPU Tesla K80 de sus servidores gratis durante unas horas al día, gracias a esto se puede acortar mucho los tiempos de entreno.

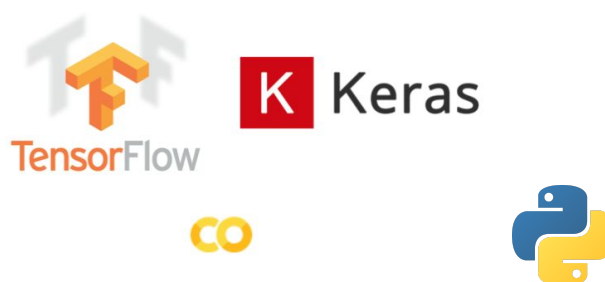


Figura 3: frameworks utilizados

Clasificación usando absorbancia

Este conjunto de datos proviene de un caso práctico en el cual se emplea el analizador VisumPalm de IRIS Technology Solutions para el control de calidad en la recepción de materia prima en la industria del plástico. Concretamente, el objetivo es identificar diferentes tipos de talco ya que en ocasiones existe el riesgo de adulteración accidental, con el consecuente impacto negativo en la calidad del producto final. Es un caso especialmente difícil, pues los espectros NIR de los minerales no suelen mostrar "formas" características acusadas, como sí ocurre con los de los compuestos orgánicos.

En la siguiente figura se muestran todas las absorbancias del conjunto completo de 680 espectros. Existen 11 tipos de muestras diferentes a clasificar de las cuales una de ellas es el mismo tipo de talco pero variando su compactación.

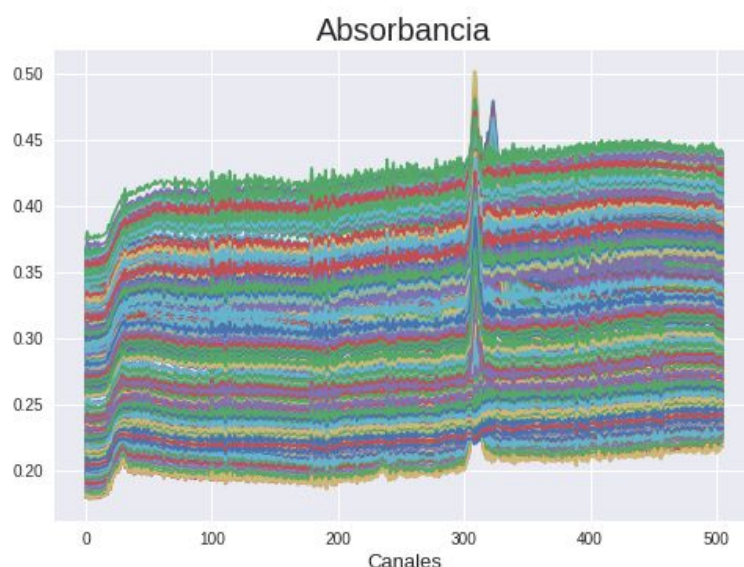
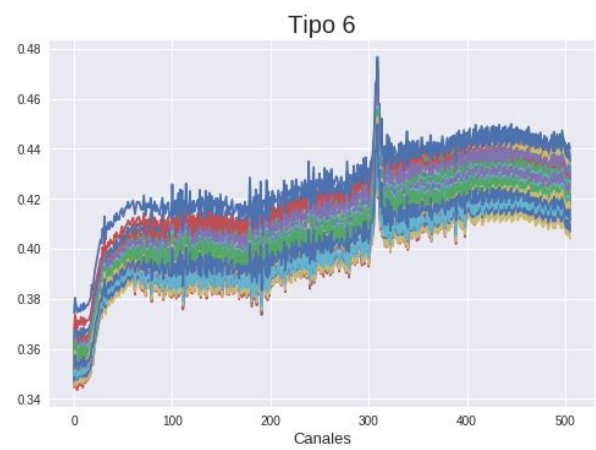
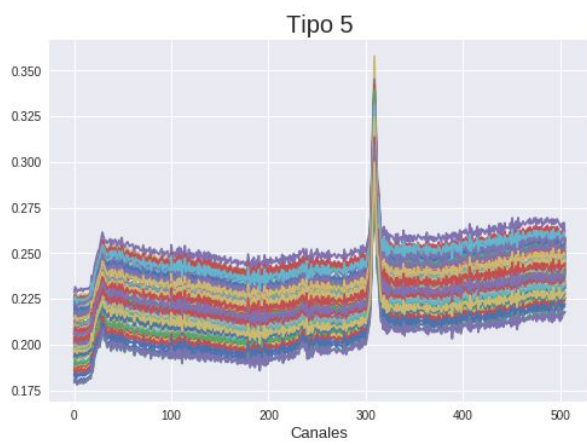
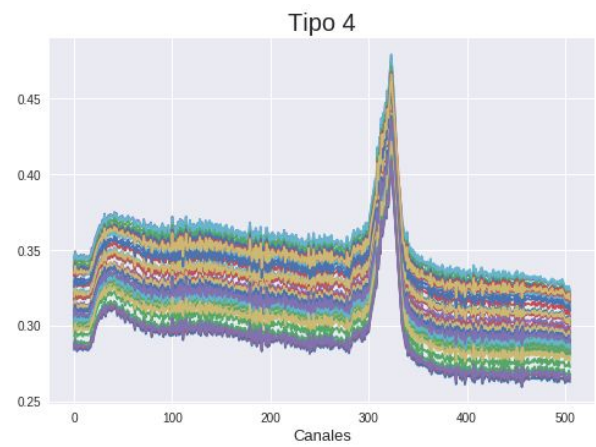
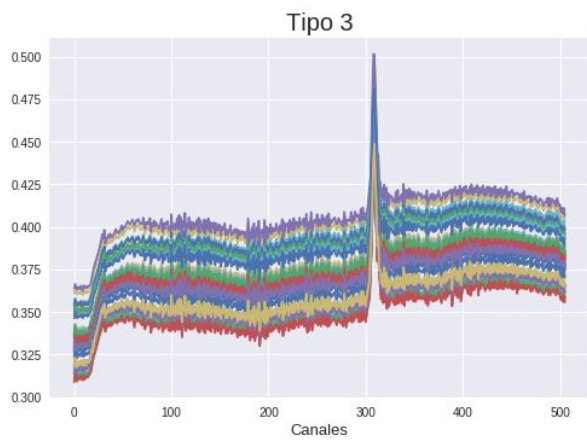
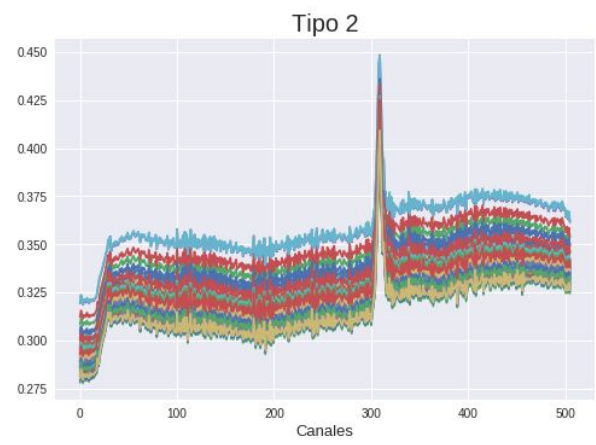
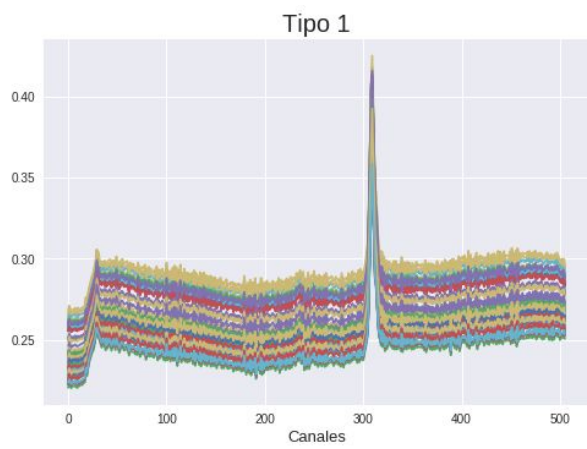


Figura 4: conjunto completo de absorbancias

A continuación se muestran los diferentes tipos por separado, a simple vista se puede ver que a excepción de tipo 4 y 9 el resto tienen un pico acentuado en la misma región. Como se ha comentado anteriormente tipo 6, 7 y 8 son el mismo compuesto químico lo único que varía su grado de compactación por lo que añade un grado de dificultad mayor para los métodos de clasificación tradicionales.



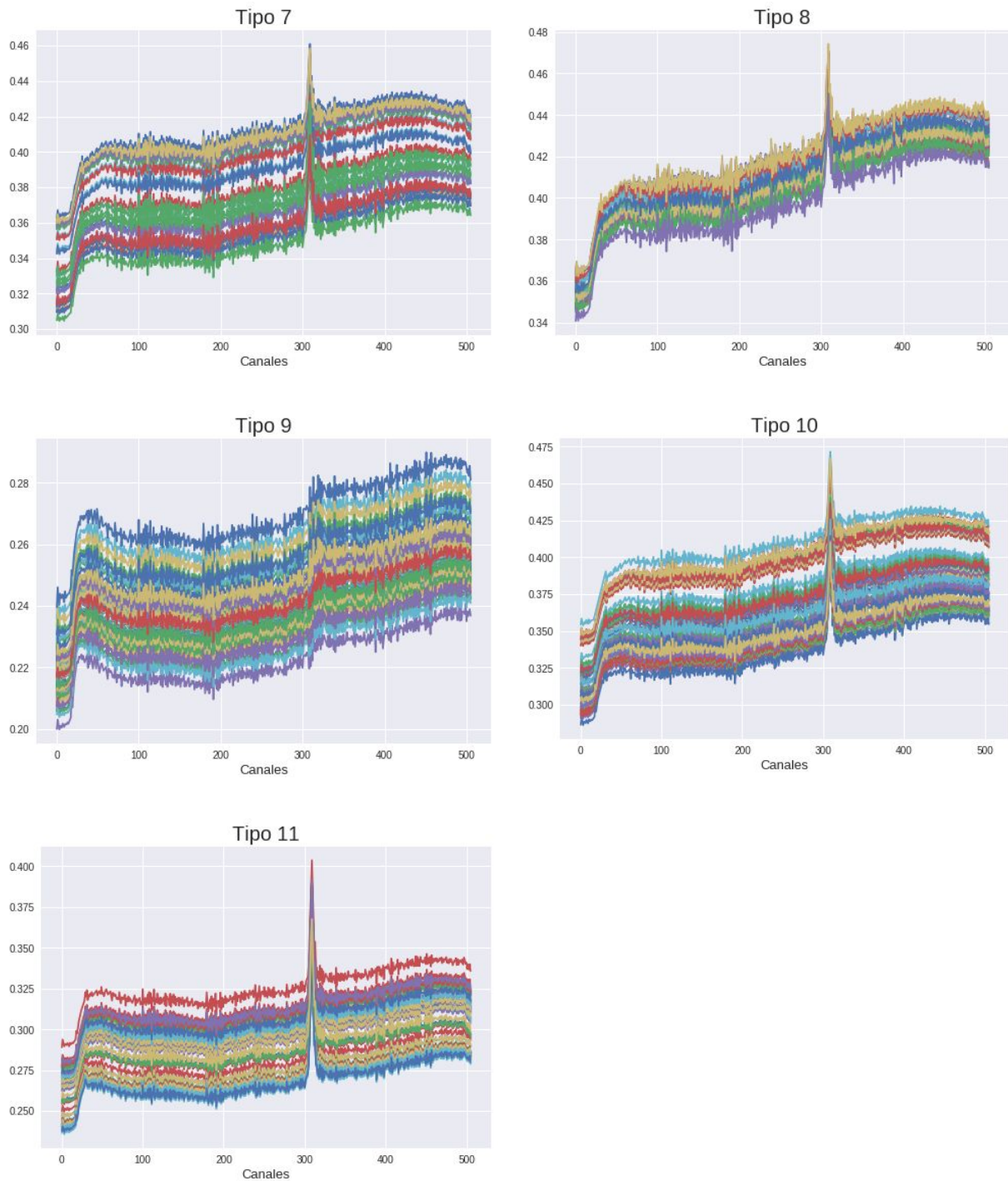


Tabla 1: figuras mostrando todos los tipos de compuestos a clasificar

Preprocesado

Los preprocesados aplicados a los datos tanto para clasificar como para cuantificar han sido a normalización y el escalado mediante la librería SciKit. Usándolos en conjunto y en este orden es cuando se ha conseguido aumentar la exactitud considerablemente reduciendo el tiempo de entreno. Esto es debido a que en redes neuronales al utilizar metodos de optimizacion como Stochastic Gradient Descent el escalado propicia a que el descenso del gradiente sea más rápido que cuando los rangos son más

amplios y la diferencia entre variables es poco homogénea. Normalizando cada espectro se consigue que las diferencias entre los del mismo tipo disminuyan y se empiece a ver a simple vista diferentes grupos de espectros. Luego con el escalado llevamos todos los datos a un mismo rango y se exaltan las diferencias de cada grupo. Estos preprocesados se han aplicado al conjunto de training y al de test por separado.

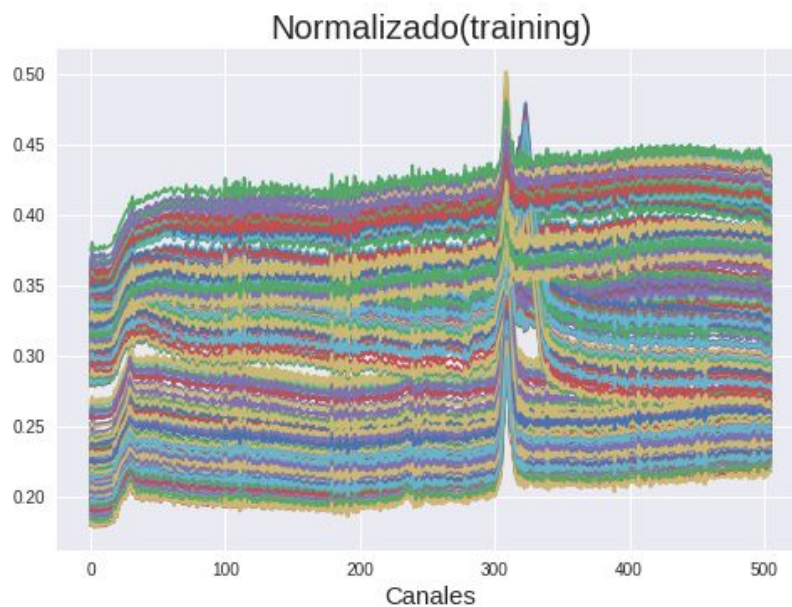


Figura 5: Conjunto de absorancias después de aplicar la normalización

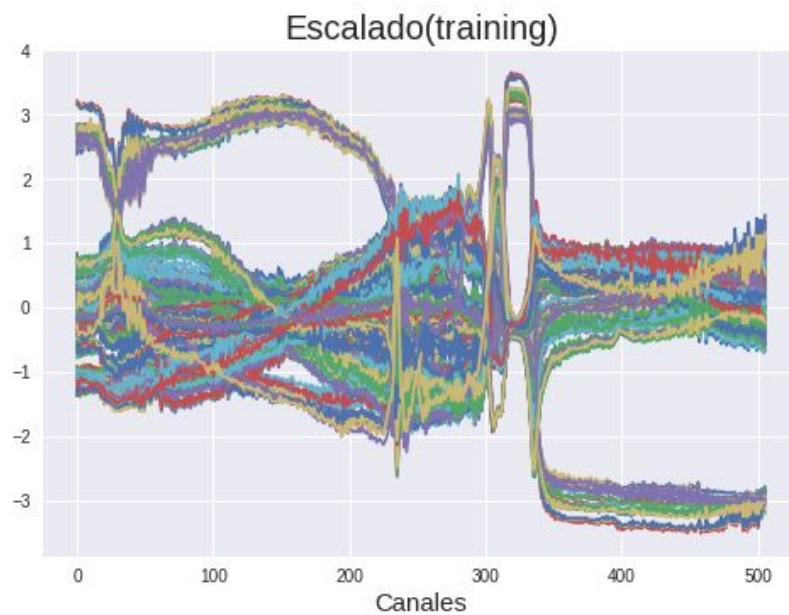


Figura 6: Conjunto de datos después de aplicar la normalización y el escalado

Red neuronal densamente conectada

Multi Layer Perceptron o MLP son tipos de redes neuronales que normalmente se usan para clasificación, específicamente cuando las clases son exclusivas. En este caso la capa de salida es una función softmax en la cual la salida de cada neurona corresponde con la probabilidad estimada de la clase correspondiente [12,13]. La función softmax se caracteriza por ser una generalización de la Función logística. Se emplea para "comprimir" un vector K-dimensional, de valores reales arbitrarios en un vector K-dimensional, de valores reales en el rango $[0, 1]$.

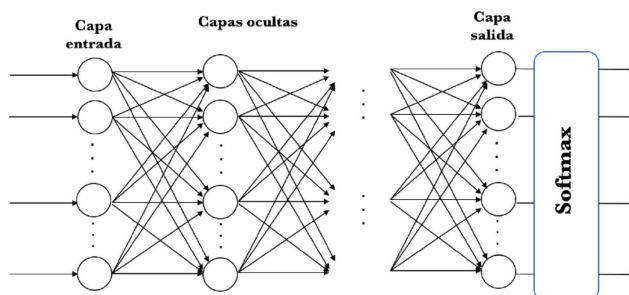


Figura 7: forma típica de una red MLP para clasificación [12]

En este primer intento se ha optado por una arquitectura muy simple con una capa de entrada y una de salida que aplica Softmax

Layer (type)	Output Shape	Param #
=====		
dense_2 (Dense)	multiple	256542
=====		
dense_3 (Dense)	multiple	6084
=====		
Total params: 262,626		
Trainable params: 262,626		
Non-trainable params: 0		
=====		
Test accuracy: 99.5%		

Al entrenar el modelo hay que considerar varios parámetros. Está es el batch size, que define el número de ejemplos del conjunto de datos de entreno que pasan hacia delante y hacia atrás por la red, cuanto mayor es el batch size más memoria se necesita. Luego están las epochs que definen cuántas veces todos los datos pasarán por la red neuronal. A mayor número de epochs, más tardará en acabar de entrenar la red sin garantizar que se obtengan mejores resultados. Incrementar las epochs puede generar overfitting, es decir que la red aprenda perfectamente el set de entreno pero luego al exponerla a datos de test, que no ha visto nunca, falle la predicción. Mientras se entrena se puede observar si disminuye el valor de *loss*, este es la suma de los errores obtenidos en cada ejemplo con los datos de training. El objetivo es tratar de minimizar este valor mientras se entrena el modelo, cuanto más pequeño es más cerca está el modelo de las predicciones correctas. La función *loss* elegida ha sido *sparec_categorical_crossentropy* [15], y el optimizador usado para buscar entre los weights de la red ha sido “Adam”.

Con una red bastante simple, a partir de 350 epochs se consigue que los datos de entreno, se obtenga una exactitud del 99.5% con los datos de test.

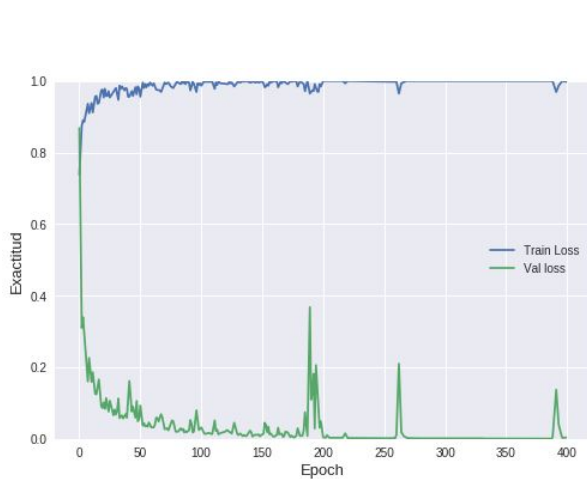


Figura 8: evolución de la exactitud según las epoch

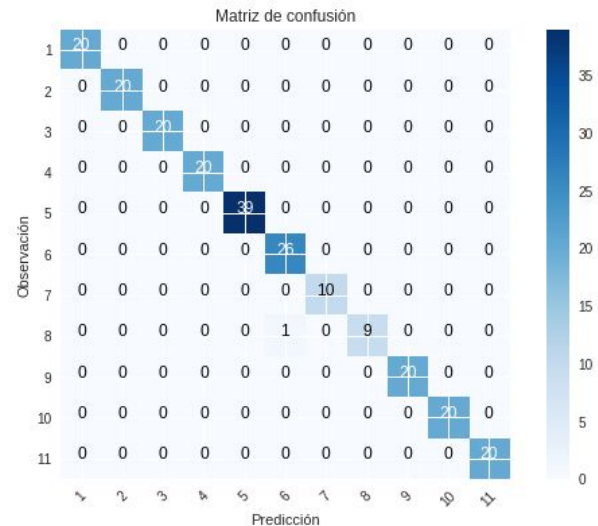


Figura 9: matriz de confusión con 99.5% de acierto

Red convolucional

Una CNN, del inglés Convolutional Neural Network, normalmente se compone de una pila de módulos convolucionales que realizan la extracción de características. Cada módulo consta de una capa convolucional seguida de una capa de agrupación. Una capa convolucional tiene el propósito principal de detectar características visuales en las imágenes. Una característica importante es que las capas convolucionales pueden aprender jerarquías espaciales de patrones al preservar las relaciones espaciales[14].

La operación de agrupación, pooling layer en inglés, simplifica la información recopilada por la capa convolucional, y crea una versión condensada de la información contenida en ella. El último módulo de una arquitectura de red convolucional es formado por una o más capas densamente conectadas que realizan la clasificación. La capa densa final en una CNN, contiene un solo nodo para cada clase objetivo en el modelo. Esta última capa usa la función de activación softmax para generar un valor entre 0 y 1 para cada nodo. Podemos interpretar los valores de softmax para una imagen dada como medidas relativas de la probabilidad de que la imagen caiga dentro de cada clase objetivo.

Las redes neuronales convolucionales 1D se pueden usar para extraer subsecuencias 1D locales de las originales y pueden identificar patrones locales dentro de la ventana de convolución[16]. Debido a que la misma transformación se aplica a cada bloque identificado por la ventana, un patrón aprendido en una posición también se puede reconocer en una posición diferente, haciendo invariante la conversión de las redes 1D. Para evitar el sobreajuste se usa Dropout, un método de Keras para aplicar regularización durante el entrenamiento. Cada activación o punto de entrada de datos, se elimina aleatoriamente del vector introducido en la capa posterior con una probabilidad ajustable “la tasa de abandono”[17]. El efecto es una interrupción de la interdependencia de las neuronas, por lo que cada neurona no puede hacer predicciones sobre patrones muy específicos de la entrada. Esto limita la capacidad de sobreajuste y da una mejor generalización para la predicción de nuevas muestras.

La arquitectura de red convolucional 1D usada es la siguiente:

Layer (type)	Output Shape	Param #
=====		
conv1d_6 (Conv1D)	(None, 497, 100)	1100
conv1d_7 (Conv1D)	(None, 488, 100)	100100
max_pooling1d_2 (MaxPooling1	(None, 162, 100)	0
conv1d_8 (Conv1D)	(None, 153, 160)	160160
conv1d_9 (Conv1D)	(None, 144, 160)	256160
global_average_pooling1d_2 ((None, 160)	0
dropout_5 (Dropout)	(None, 160)	0
dense_7 (Dense)	(None, 12)	1932
=====		
Total params: 519,452		
Trainable params: 519,452		
Non-trainable params: 0		
=====		
Test accuracy: 99.7%		

Como muestran las siguientes figuras se consigue mejorar ligeramente la exactitud gastando el mismo número de epochs, llegando así al 99.7% .

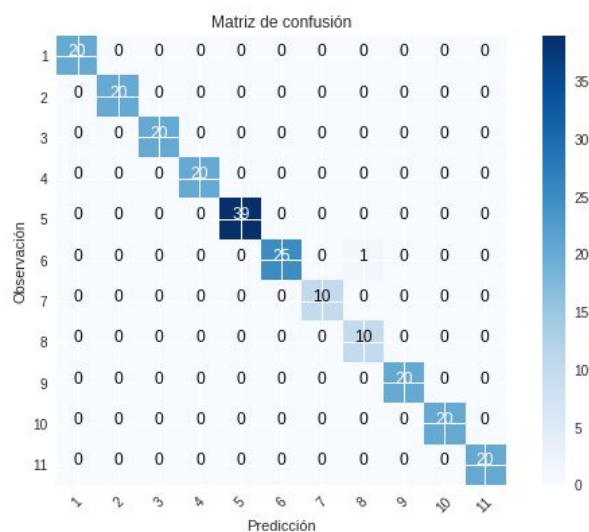
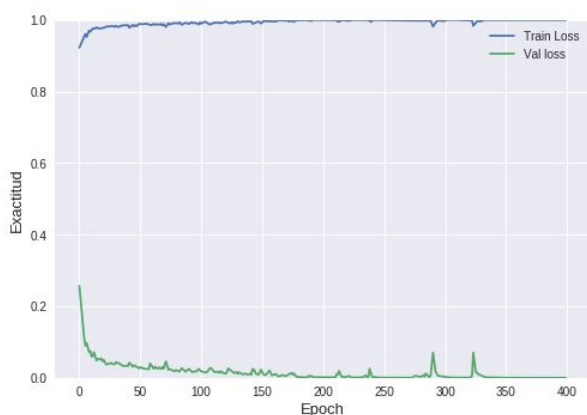


Figura 10: evolución de la exactitud con los datos de training y la función de *loss* según aumentan las epoch

Figura 11: matriz de confusión clases predichas vs clases observadas

Clasificación usando raw data

Uno de los objetivos es comprobar si las arquitecturas de redes neuronales con varias capas, pueden extraer información necesaria incluso con los datos raw. Los datos raw presentan irregularidades que principalmente se deben a dos factores: uno estocástico, que proviene del ruido de cada canal del sensor, y otro "fijo" debido a la diferencia de respuesta de un canal respecto a otro.

A continuación se muestran los espectros de los mismos 11 tipos de compuestos analizados anteriormente en su versión raw. Estos mismos raw han sido usados para generar las absorbancias del caso anterior por lo que estamos ante el mismo experimento.

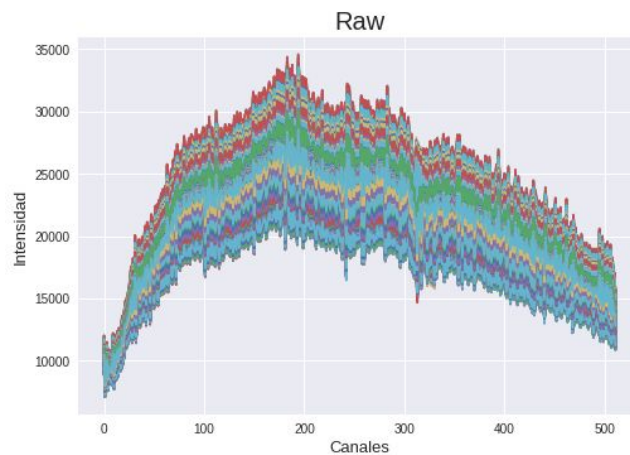


Figura 12: conjunto de espectros raw de todos los tipos de muestra

Preprocesado

Se han aplicado los mismos preprocesados, normalización y escalado al conjuntos de training y al de test. Se puede observar que la señal al ser más ruidosa originariamente al aplicar el escalado se acentúa más.

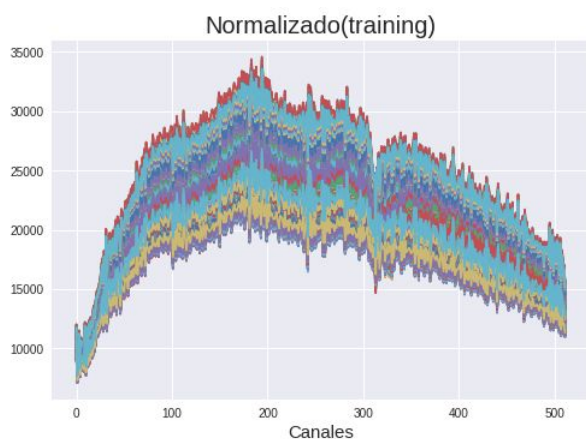


Figura 13: raw normalizado

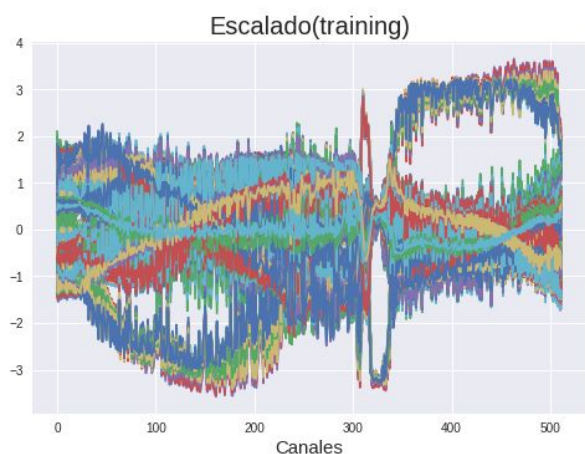


Figura 14: raw normalizado y escalado

Red neuronal densamente conectada

La arquitectura de red densamente conectada ha sido la misma que para las absorbancias y aunque en entrenamiento ha llegado a la tasa de 100% de exactitud, no lo ha conseguido con los datos de test. Esta red tiene problemas con las clases 6 y 8, algo esperable puesto que son las que más complejidad tienen al ser el mismo compuesto químico pero con diferente compactación. Aumentando el número de epochs o añadiendo más capas intermedias no se ha conseguido mejorar la tasa de acierto.

Layer (type)	Output Shape	Param #
dense (Dense)	multiple	262656
dense_1 (Dense)	multiple	6156
Total params: 268,812		
Trainable params: 268,812		
Non-trainable params: 0		
Test accuracy: 0.9912280701754386		

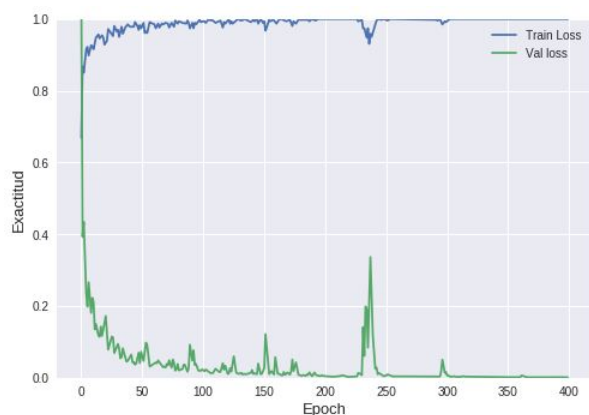


Figura 15: evolución de la exactitud en los datos de training y la función de *loss* según aumentan las epoch

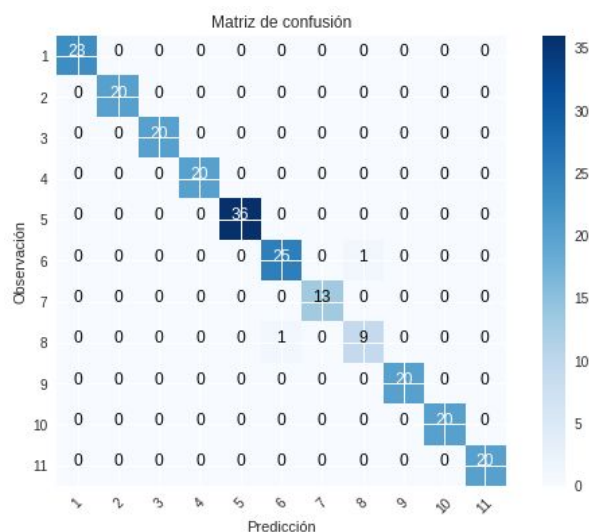


Figura 16: matriz de confusión clases predichas vs clases observadas

Red convolucional

Usando la misma red convolucional que para las absorbancias, se ha conseguido mejor resultado que con la anterior, una exactitud del 99.93%. La arquitectura de esta red es más compleja, requiere de más parámetros a entrenar e incluye una capa de regularización por o que el sobreajuste en datos de training está más controlado.

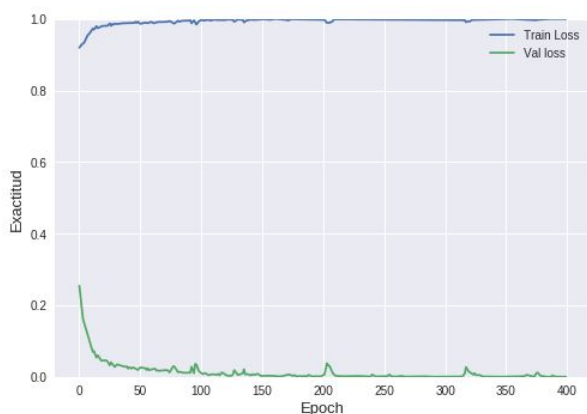


Figura 17: evolución de la exactitud en los datos de training y la función de *loss* según aumentan las epoch

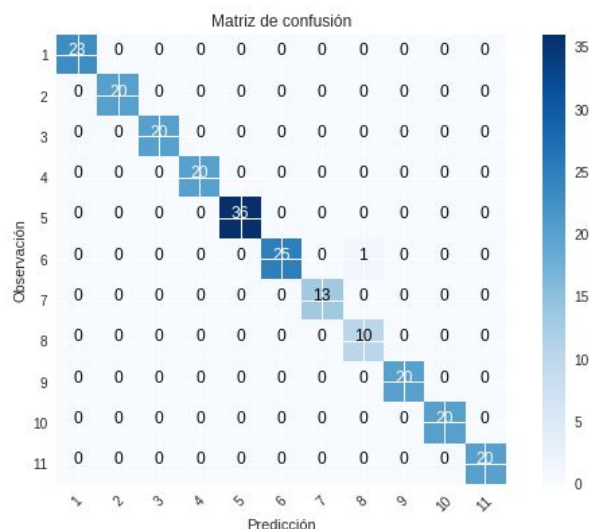


Figura 18: matriz de confusión clases predichas vs clases observadas

Comparativa PLS-DA vs Redes neuronales artificiales

Es una práctica habitual en quimiometría trabajar con absorbancias, pues dicho enfoque es compatible con el modo de adquirir los espectros en la mayoría de los casos. Esto es, cuando se dan las condiciones para que se cumpla la ley de Lambert-Beer, ley que relaciona la absorción de luz con las propiedades del material atravesado. Por ello solo se muestra la matriz de confusión con los resultado de predecir con un modelo PLS-DA a partir de las absorbancias. Se puede observar que este tipo de modelo es incapaz de distinguir entre tantas clases tan homogéneas, por lo que se queda en un 44% de exactitud a diferencia del 99.93% conseguido por las redes convolucionales.

Actual Class												
	1	2	3	4	5	6	7	8	9	10	11	
Predicted	1	0	0	0	0	0	0	0	0	0	0	0
Predicted	2	0	0	0	0	0	0	0	0	0	0	0
Predicted	3	0	0	25	0	0	0	0	0	0	0	0
Predicted	4	0	0	0	18	0	0	0	0	0	0	0
Predicted	5	0	0	0	0	13	0	0	0	0	0	0
Predicted	6	0	0	0	0	0	0	0	0	0	0	0
Predicted	7	0	0	0	0	0	0	5	0	0	0	0
Predicted	8	0	0	0	0	0	0	0	0	0	0	0
Predicted	9	0	0	0	0	0	0	0	25	0	0	0
Predicted	10	0	0	0	0	0	0	0	0	9	0	0
Predicted	11	0	0	0	0	0	0	0	0	0	5	0
Unassigned		18	23	0	0	26	27	3	9	2	3	20

Tabla 2: clases predichas comparadas con las observaciones con PLS-DA

A modo resumen se presenta la siguiente tabla con los resultados en exactitud de los diferentes modelos según sus tipos de datos. Estas métricas se corresponden al mejor resultado de tres iteraciones de mezclar y repartir los datos entre test y train, generar modelo y hacerlo predecir. Para hacer la tabla más compacta se ha usado DNN como sustituto a Red neuronal densamente conecta y CNN para referirnos a red convolucional.

	Absorbancia			Raw	
	DNN	CNN	PLS-DA	DNN	CNN
Exactitud:	99.5%	99.7%	44%	99.1%	99.93%

Tabla 3: métricas de exactitud para cada tipo de modelo

En la tabla se demuestra que las redes neuronales son ampliamente superiores al método analítico PLS-DA. La red convolucional con datos raw es la más prometedora puesto que cumple otro de los objetivos, el de comprobar si la red es capaz de distinguir correctamente los espectros sin necesidad de calibración.

Cuantificación usando absorbancias

El siguiente conjunto de datos proviene de otro caso práctico usando el mismo tipo de analizador. El objetivo es cuantificar varios parámetros de calidad en la carne picada antes de ser usada para embutidos. El conjunto de datos a tratar contiene varios grupos de espectros tomado a distintos porcentajes de humedad en la carne. En la siguiente figura se muestran las 780 las absorbancias del conjunto completo de datos.

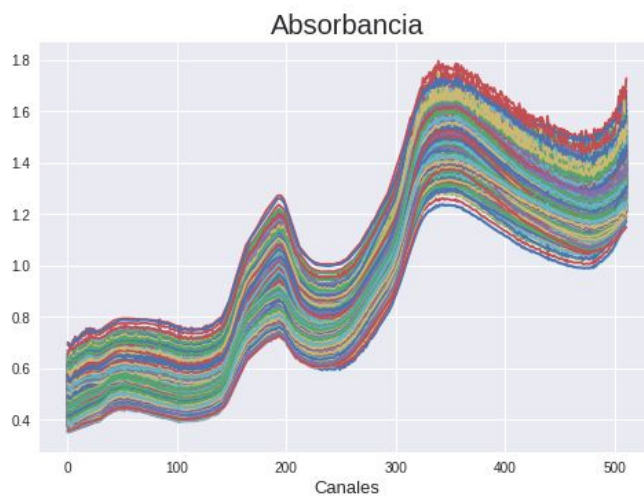
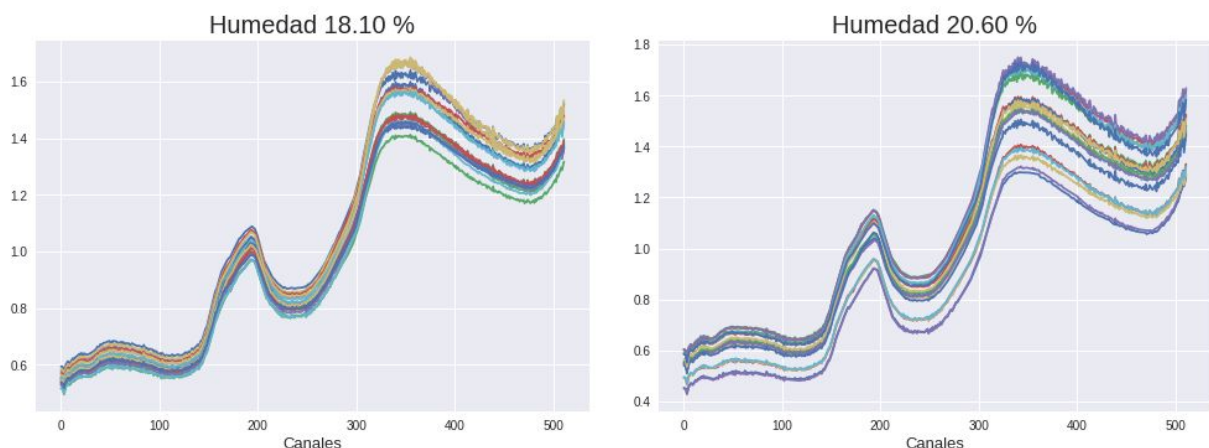


Figura 19: conjunto de absorbancias de carne picada

Agrupando los datos por porcentaje de humedad se puede observar como los espectros pueden llegar a ser muy distintos. Existen tres factores principales, el primero es que a un mismo nivel de humedad, hay varios niveles de grasa y proteína; el segundo, que los espectros están afectados por el scattering y el tercero, porque cada espectro corresponde a una región de muestreo de unos 10 mm sobre una muestra de carne picada bastante heterogénea. Debido a que existen tantas fuentes de variabilidad se justifica la necesidad de disponer herramientas más potentes para el datamining.



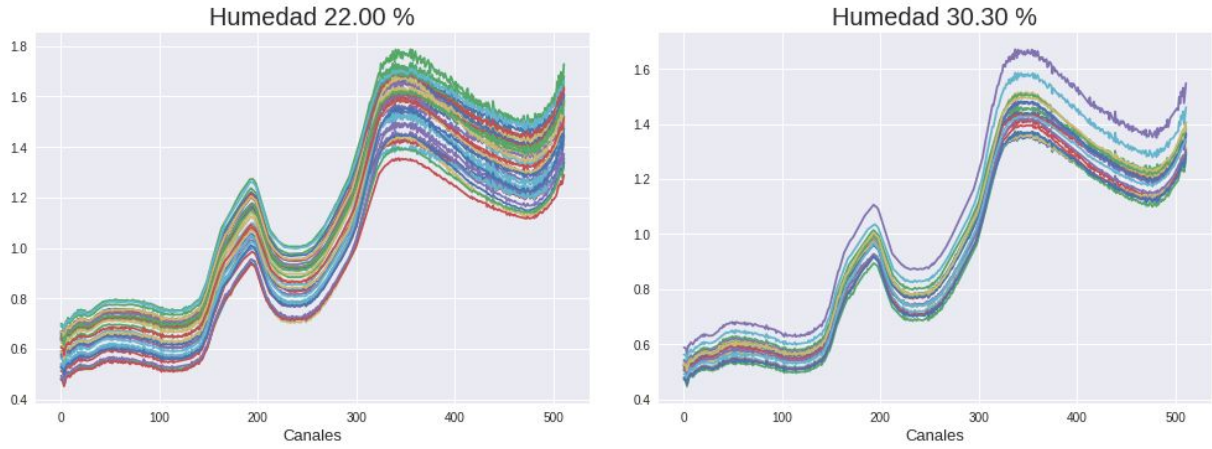


Tabla 4: figuras mostrando los subconjuntos de absorbancias según el porcentaje de humedad de la muestra

A diferencia de la clasificación, que los preprocesados fueron los mismos para cada tipo de red, en cuantificación se han ido tratando diferentes según el resultado obtenido. De la misma forma que con el experimento anterior, el conjunto de datos se ha separado en un tercio para test y el resto para training. Se ha usado la misma herramienta, *StratifiedShuffleSplit*, para obtener subconjuntos de datos aleatorios que conservan el porcentaje para cada subconjunto con la misma humedad. Las métricas usadas para comparar los diferentes modelos son Root-Mean-Squared-Error y Mean Absolute Error, ambos usados para medir la efectividad del modelo al predecir con los datos de test[18]. Para la regresión se ha usado el optimizador RMSProp. Los optimizadores son usado para encontrar un mínimo global en el que la función de *loss* tiene el valor mínimo posible. RMSProp es la versión mejorada de Stochastic Gradient Descent, y se a usado conjunto con la métrica de referencia Mean-Squared-Error.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_{pred,i} - y_i)^2}{n}} \quad MAE = \frac{\sum_{i=1}^n |y_{pred,i} - y_i|}{n} \quad MSE = \frac{\sum_{i=1}^n (y_{pred,i} - y_i)^2}{n}$$

Red neuronal densamente conectada

Para este tipo de red, normalizar y escalar los datos ha sido el conjunto de preprocesados que mejor resultado ha conseguido. La arquitectura de esta red contiene más capas de 64 neuronas cada una. Usando una sola capa como en clasificación no es suficiente para obtener resultados satisfactorios en regresión. El learning rate configurado para el optimizador ha sido 0.001, este parámetro controla cuánto se ajustan los weights de la red con respecto a la loss del gradiente.

Layer (type)	Output Shape	Param #
dense_37 (Dense)	(None, 512)	262656
dense_38 (Dense)	(None, 256)	131328
dense_39 (Dense)	(None, 64)	16448
dense_40 (Dense)	(None, 64)	4160
dense_41 (Dense)	(None, 64)	4160
dense_42 (Dense)	(None, 64)	4160
dense_43 (Dense)	(None, 64)	4160
dropout_4 (Dropout)	(None, 64)	0
dense_44 (Dense)	(None, 1)	65
Total params: 427,137		
Trainable params: 427,137		
Non-trainable params: 0		
Test MAE = 0.65		
Test RMSE = 1.36		

Con este tipo de red se han conseguido los valores MAE y RMSE, muy bajos sin necesidad de usar muchas epochs. Aunque existen predicciones con errores de hasta ± 7 la mayor parte entra en el rango de ± 1 .

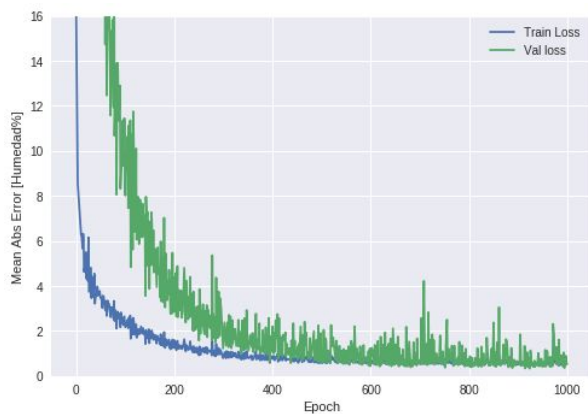


Figura 20: Evolución del MAE con los datos de training y loss mostrando MSE, según pasan las epoch

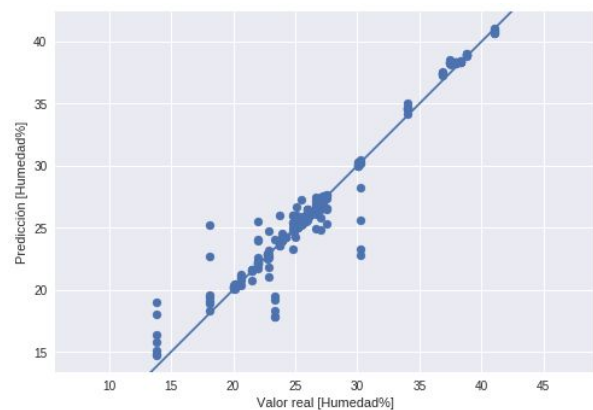


Figura 21: predicciones contra observaciones

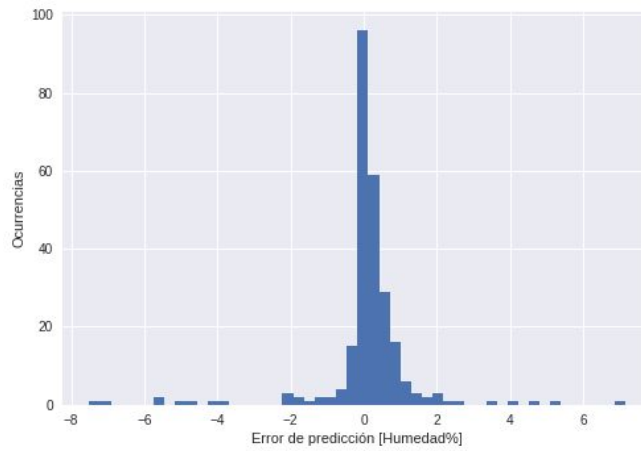


Figura 22: frecuencia del error de predicción

Red convolucional

El ruido estocástico puede reducirse mediante suavizado Savitzky-Golay[19]. La idea base es que un espectro de materia condensada, todo lo que no es gas, con poco ruido debería ser típicamente suave, ya que se debe a la superposición de muchos armónicos espectrales provenientes de vibraciones fundamentales que son más bien anchas. Este pretratamiento no suele ser crítico cuando se trata de clasificar, siempre y cuando los espectros de cada clase tengan una forma característica, pero ayuda mucho si se pretende cuantificar. Para este tipo de red se ha probado aplicar un filtro cuadrático Savitzky-Golay con ventana de 9 canales, seguidamente una normalización y un escalado tal y como se muestran en las siguientes figuras.

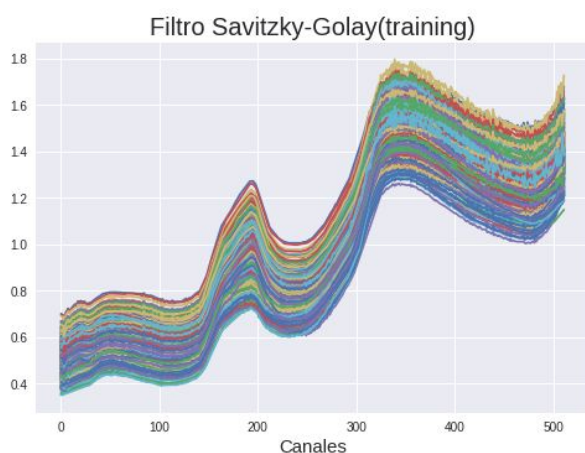


Figura 23: filtro de alisado aplicado

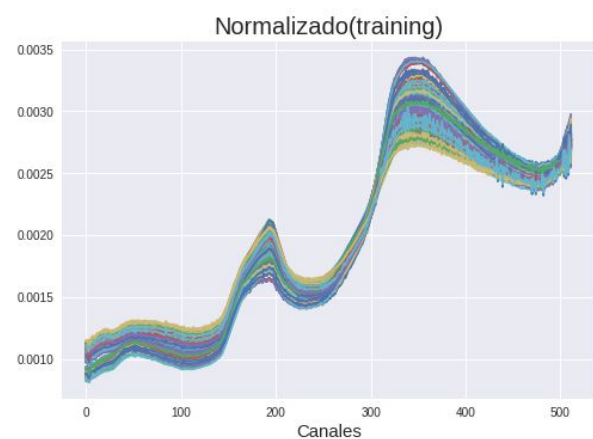


Figura 24: espectros normalizados

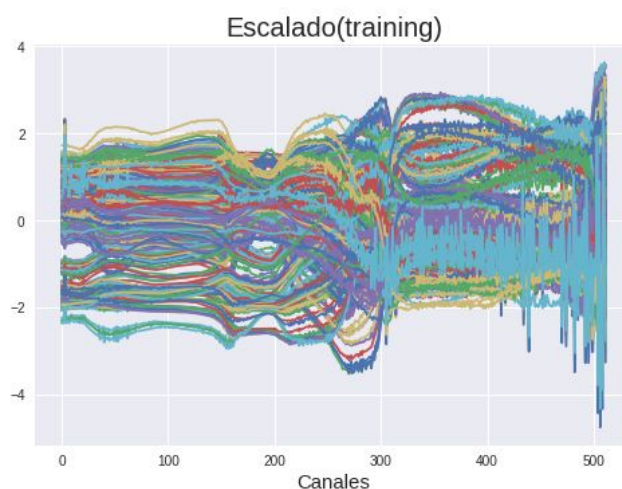


Figura 25: escalado del conjunto de datos

La arquitectura usada es semejante a la de clasificación. Se consiguen resultados cercanos a las redes densamente conectadas a partir de las 6000 epochs, aun así la dispersión es mayor y encontramos que muchas predicciones caen en un error de ± 4 .

Layer (type)	Output Shape	Param #
=====		
conv1d_37 (Conv1D)	(None, 503, 128)	1408
conv1d_38 (Conv1D)	(None, 494, 128)	163968
max_pooling1d_10 (MaxPooling)	(None, 164, 128)	0
conv1d_39 (Conv1D)	(None, 155, 160)	204960
conv1d_40 (Conv1D)	(None, 146, 160)	256160
global_average_pooling1d_10	(None, 160)	0
dropout_10 (Dropout)	(None, 160)	0
dense_10 (Dense)	(None, 1)	161
=====		
Total params: 626,657		
Trainable params: 626,657		
Non-trainable params: 0		
=====		
Test MAE = 1.06		
Test RMSE = 1.87		

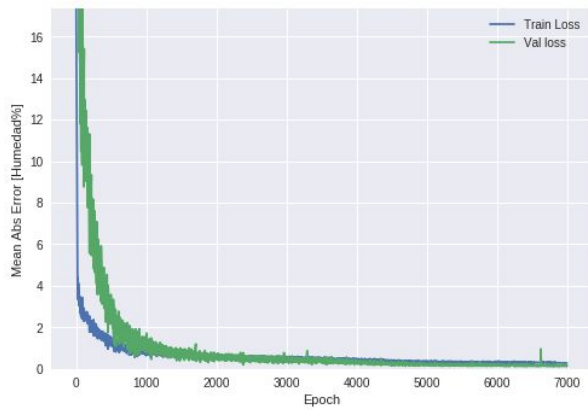


Figura 26: Evolución del MAE con los datos de training y *loss* mostrando MSE, según pasan las epoch

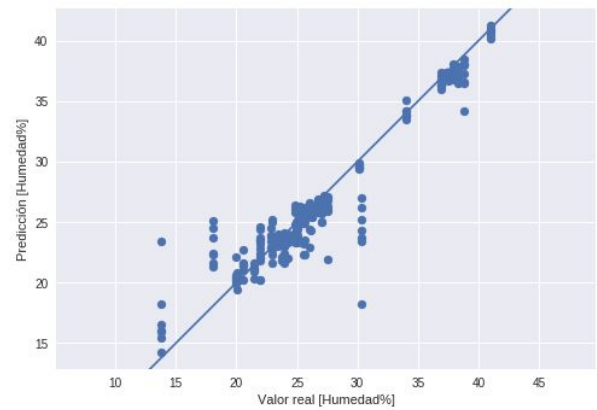


Figura 27: predicciones contra observaciones

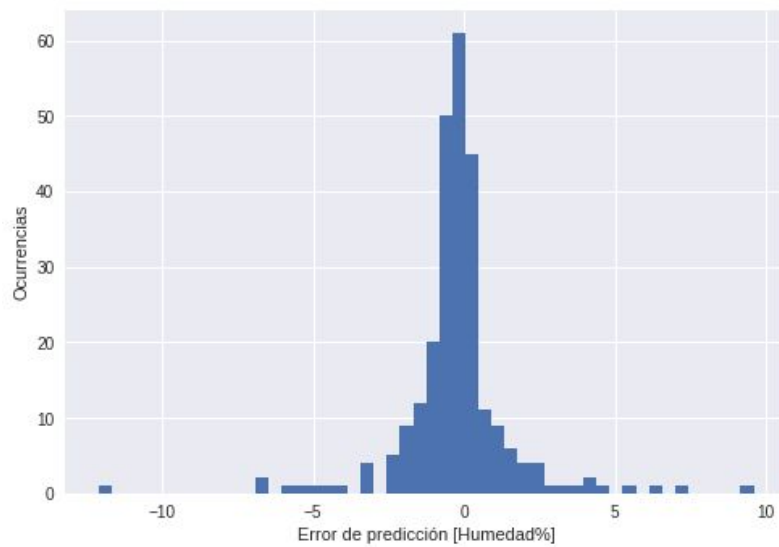


Figura 28: frecuencia del error de predicción

Cuantificación usando raw data

Red neuronal densamente conectada

Para este tipo de red primero se ha probado aplicar un filtro cuadrático Savitzky-Golay con ventana de 9, seguidamente una normalización y un escalado tal y como se muestran en las siguientes figuras.

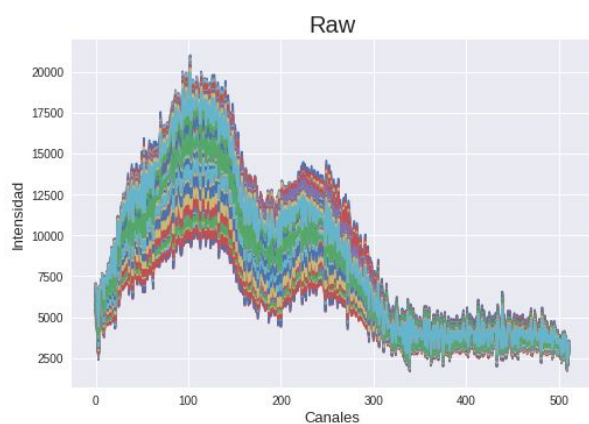


Figura 29: conjunto total de espectros raw

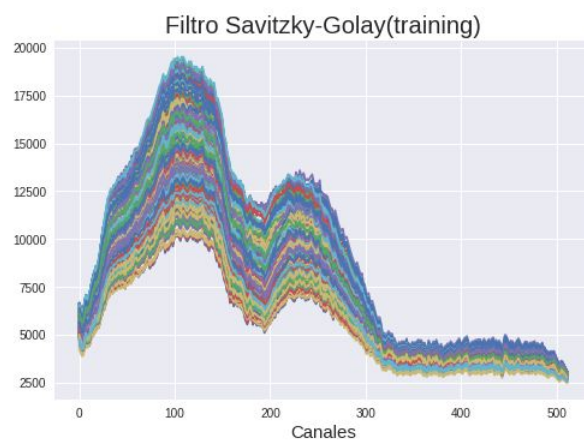


Figura 30: filtro de alisado aplicado a training

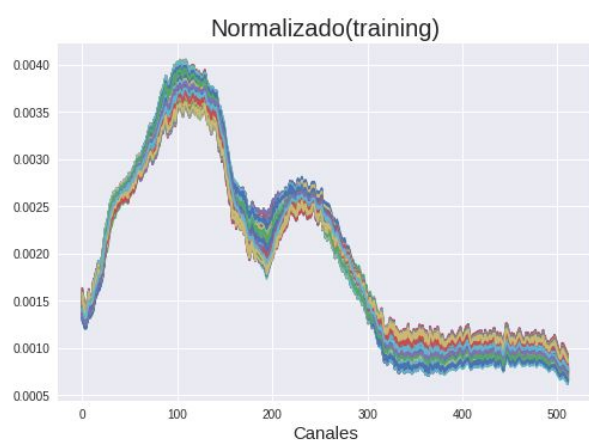


Figura 31: espectros normalizados

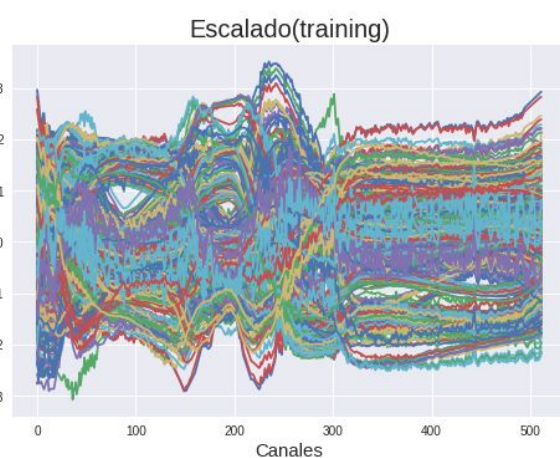


Figura 32: escalado a 0 todo el conjunto de datos

La arquitectura de red usada es similar a la utilizada con las absorbancias pero añadiendo una capa de Dropout.

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 512)	262656
dense_9 (Dense)	(None, 256)	131328
dense_10 (Dense)	(None, 64)	16448
dense_11 (Dense)	(None, 64)	4160
dense_12 (Dense)	(None, 64)	4160
dense_13 (Dense)	(None, 64)	4160
dense_14 (Dense)	(None, 64)	4160

dropout_1 (Dropout)	(None, 64)	0
dense_15 (Dense)	(None, 1)	65
=====		
Total params: 427,137		
Trainable params: 427,137		
Non-trainable params: 0		
=====		
Test MAE = 0.92		
Test RMSE = 1.91		

Aunque los valores de MAE y RMSE son superiores a su homóloga con absorbancias, la frecuencia en que las predicciones son cercanas a un error 0 es mayor usando esta red con datos raw. Por contra el abanico de predicciones totalmente fuera de rango es amplio, llegando a los -10 y $+7.5$. Sin embargo, la frecuencia de estos outliers es muy baja por lo que hace que este modelo sea interesante.

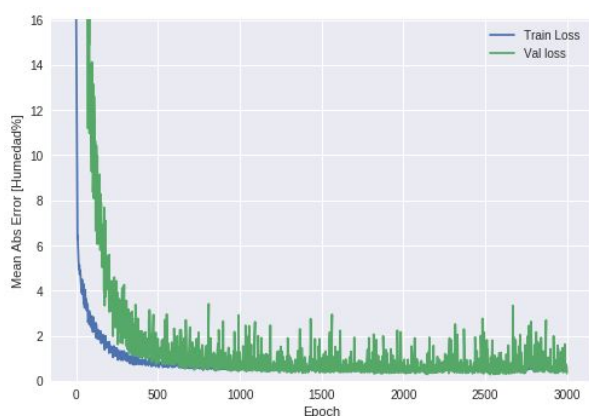


Figura 33: Evolución del MAE con los datos de training y loss mostrando MSE, según pasan las epoch

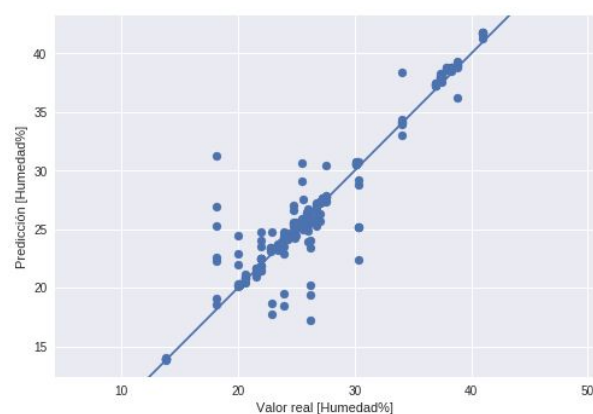


Figura 34: predicciones contra observaciones

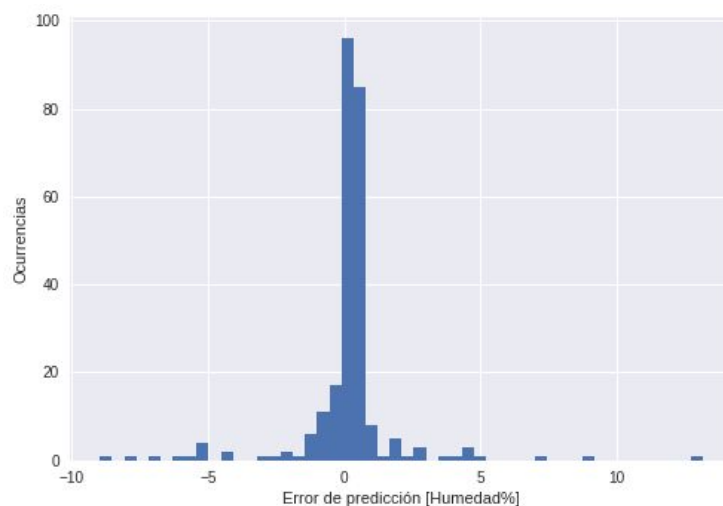


Figura 35: frecuencia del error de predicción

Para el mismo tipo de red también se ha probado a usar otro preprocesado más. Conjunto con el filtro de suavizado se ha aplicado la primera derivada a todos los espectros raw. El suavizado ha sido un filtro cuadrático Savitzky-Golay con ventana de 9, luego como en los anteriores casos una normalización y un escalado tal y como se muestran en las siguientes figuras.

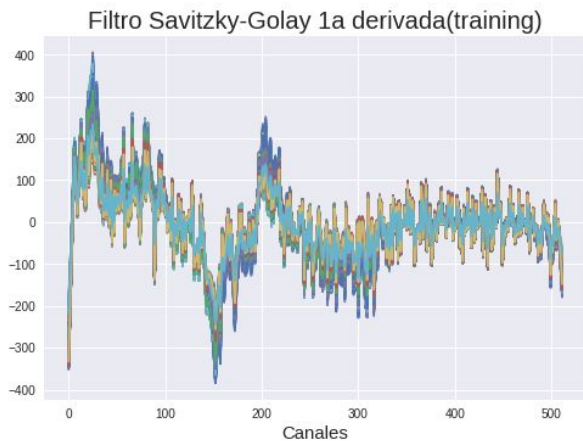


Figura 36: filtro suavizado y primera derivada a los raw

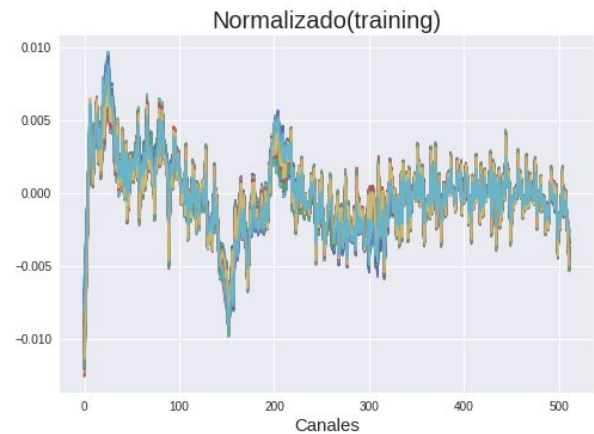


Figura 37: normalización de los raw

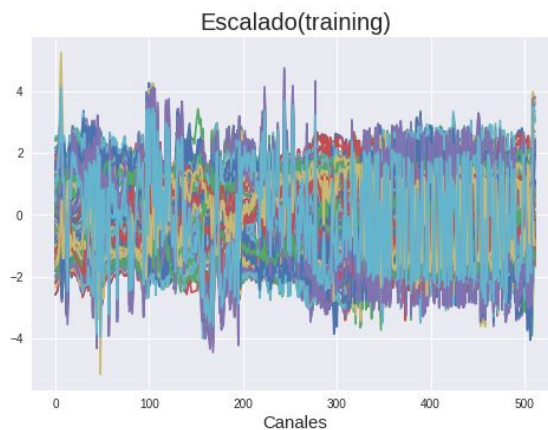


Figura 38: escalado de los espectros

Resultados :

Test MAE: 0.44

Test RMSE: 0.96

Con esta configuración se consigue con los datos de test los resultados en MAE y RMSE más bajos. El abanico de valores fuera de rango es menor, entre ± 4 . Al igual que el resto de modelos encontramos que la zona de humedad 30% las predicciones tienden a fallar bastante.

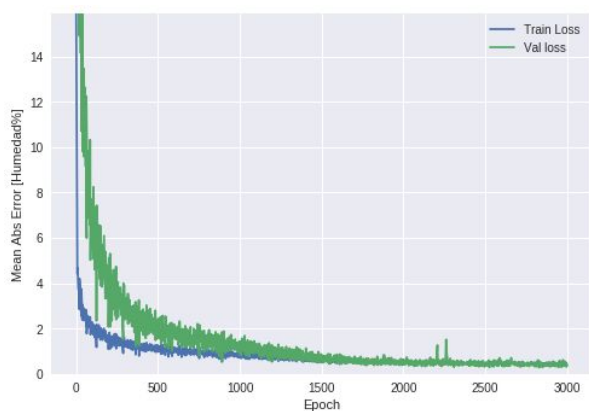


Figura 39: Evolución del MAE con los datos de training y *loss* mostrando MSE, según pasan las epoch

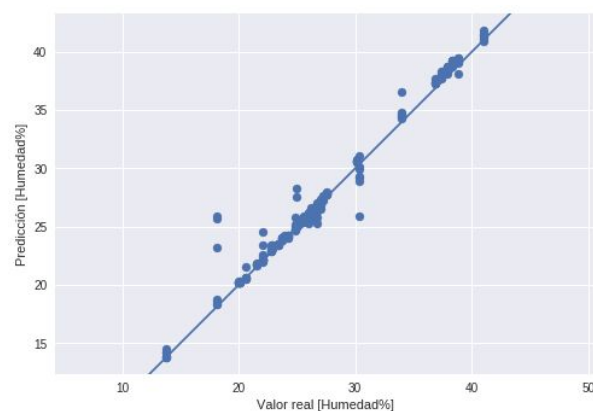


Figura 40: predicciones contra observaciones

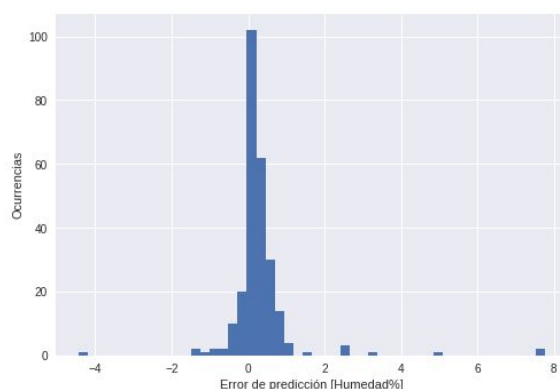


Figura 41: frecuencia del error de predicción

Red convolucional

Para la red convolucional el preprocesado que mejor a funcionado ha sido el de aplicar un filtro cuadrático Savitzky-Golay con ventana de 9, la primera derivada, seguida de una normalización y un escalado.

La métricas con los espectros raw son mejores para este tipo de red que con las absorbancias. El abanico de outliers es bajo pero de amplio error, la frecuencia de predicción está centrada en error 0 aunque hay bastantes predicciones que caen en ± 1 y un porcentaje menor entre ± 2 .

Test MAE = 0.85

Test RMSE = 1.46

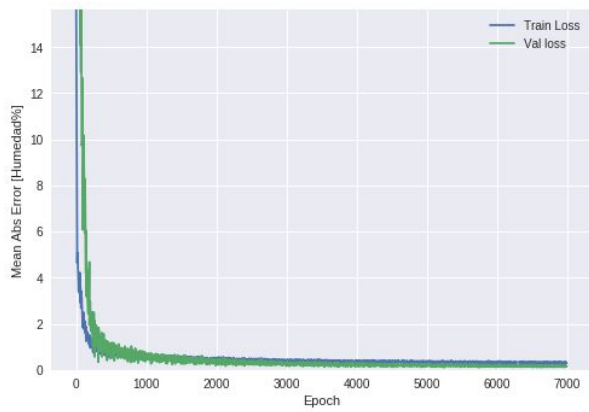


Figura 42: Evolución del MAE con los datos de training y loss mostrando MSE, según pasan las epoch

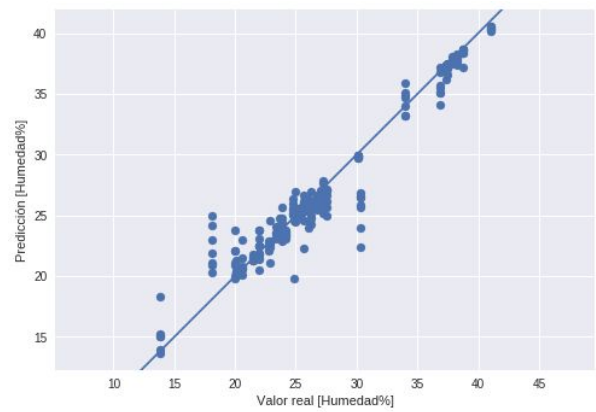


Figura 43: predicciones contra observaciones

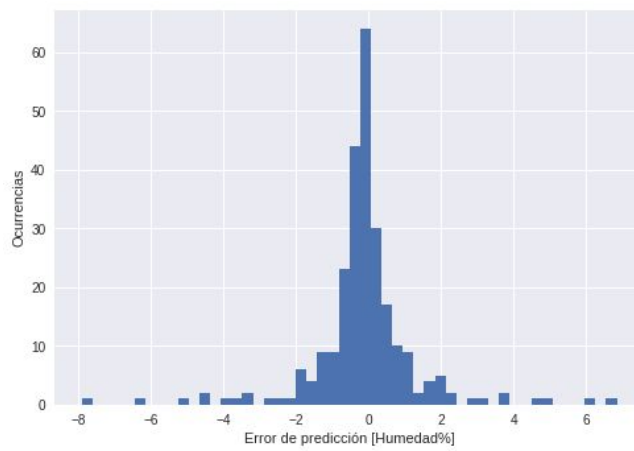


Figura 44: frecuencia del error de predicción

Comparativa PLS vs Redes neuronales artificiales

La siguiente figura muestra los valores de las predicciones en contraposición a los valores reales, para un modelo PLS creado con las mismas absorbancias presentadas. Los puntos en rojo en la gráfica son las predicciones con los datos de test, mientras que los puntos grises son las predicciones con los datos de entreno. La métrica RMSE obtenida es superior a la de los modelos obtenidos con redes neuronales artificiales. La gráfica y el resultado es cedido por lo que no se ha podido calcular la métrica MAE. RMSE = 1.9195

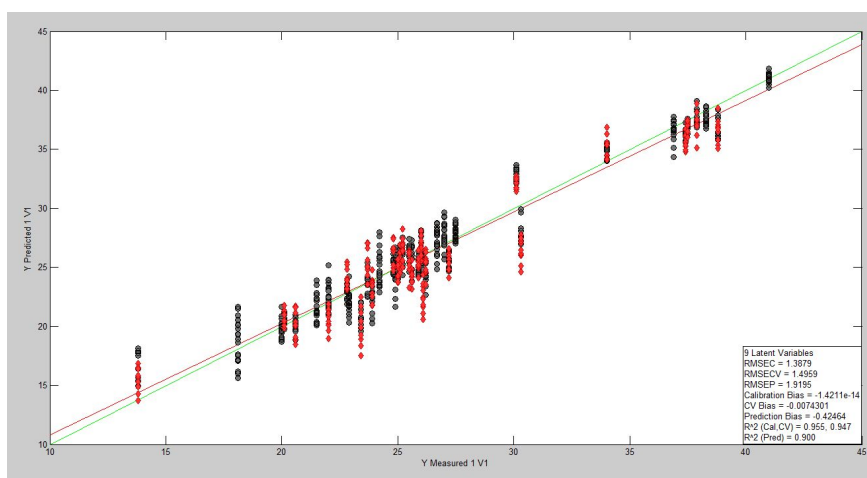


Figura 45: modelo PLS usando absorbancias

A modo resumen la siguiente tabla muestra las métricas para cada uno de los modelos mostrados. Estas métricas se corresponden al mejor resultado de tres iteraciones de mezclar y repartir los datos entre test y train, generar modelo y hacerlo predecir. Para hacer la tabla más compacta se ha usado DNN como sustituto a Red neuronal densamente conecta y CNN para referirnos a red convolucional.

	Absorbancia			Raw		
	DNN	CNN	PLS	DNN	DNN	CNN
Preprocesado:	DNN	CNN	PLS	DNN	DNN	CNN
Suavizado		X	X	X	X	X
1ª Derivada					X	X
Normalizar	X	X		X	X	X
Escalar	X	X	X	X	X	X
MAE	0.65	1.06	--	0.92	0.44	0.85
RMSE	1.36	1.87	1.91	1.91	0.96	1.46

Tabla 5: resumen de métricas obtenidas según modelo y tipo de datos

Ya sean DNN o CNN con absorbancias o espectros raw los modelos predictivos conseguidos ofrecen métricas de error más favorables que PLS con absorbancias. El objetivo de tener unas herramientas de predicción que ofrezcan mejor resultado que PLS y PLS-DA se ha cumplido. Las redes densamente conectadas han tenido un rendimiento superior a las CNN para cuantificar, especialmente con datos raw. La siguiente tabla muestra las configuraciones a elegir teniendo en cuenta algunas consideraciones.

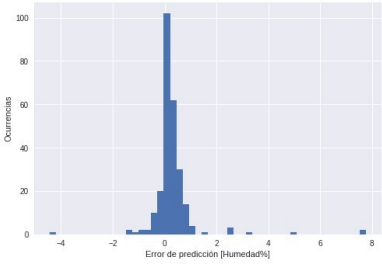
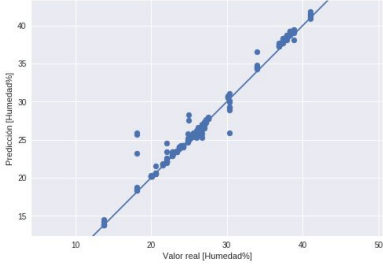
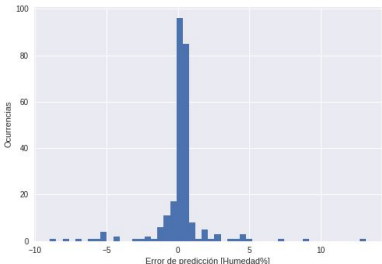
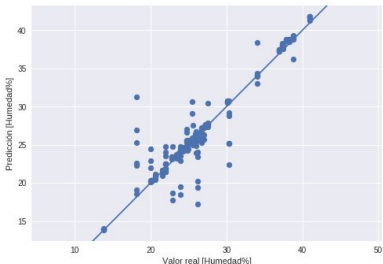
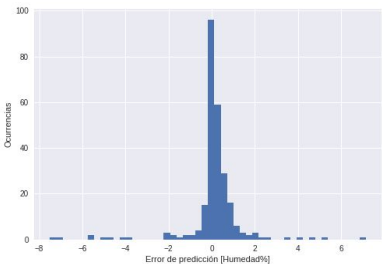
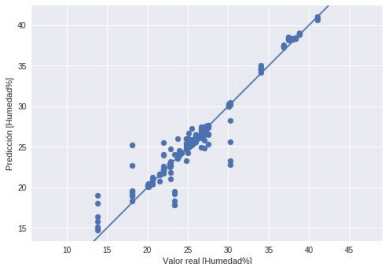
<p>DNN Raw con todos los preprocesados. Se obtiene el modelo con mejores métricas. El error está centrado en ± 0.5 con</p>		
<p>DNN Raw sin aplicar primera derivada. Modelo que en media predice muy bien con error muy cercano a 0, pero cuando predice mal el rango es de ± 12. En casos prácticos tomar 5 muestras y aplicar la mediana a las predicciones puede minimizar el problema de los outliers haciendo este modelo mucho más certero.</p>		
<p>DNN absorbancia, en el caso de necesitar usar absorbancias este sería el modelo a usar.</p>		

Tabla 6: mejores configuraciones de preprocesado, tipo de redes, y datos según resultados de predicción

Imagen hiperespectral

Descripción de datos

Los datos de este experimento han sido obtenidos con el equipo de imagen hiperespectral Hypera. Este cuenta con una par de cámaras, una NIR y la otra SWIR, del inglés Shortwave Infrared. Los datos obtenidos y tratados se corresponden a la cámara NIR. El objetivo de este experimento es el de ver si es posible detectar cuerpos extraños dentro de un producto procesado, en esta caso fiambre de jamón cocido. Para generar el conjunto de datos de entrenamiento del modelo, se tomaron varias capturas en días consecutivos de muestras de jamón cocido y muestras de jamón cocido con un material a detectar. Las muestras de jamón con material a detectar fueron en forma de sándwich, es decir capas de jamón, material a detectar, y capas de jamón. La profundidad a detectar oscila entre los 2 y 4 milímetros. Un conjunto de datos aparte ha sido adquirido para la validación, este consiste también en un tipo de sándwich con varias capas de jamón, varias muestras de material a detectar y capas de jamón encima.

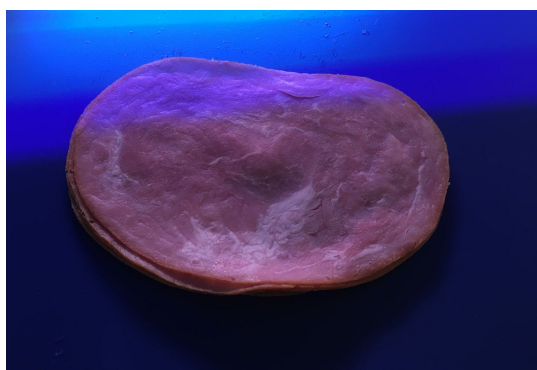


Figura 46: Muestras de jamón cocido



Figura 47: Sándwich jamón con plástico PMMA

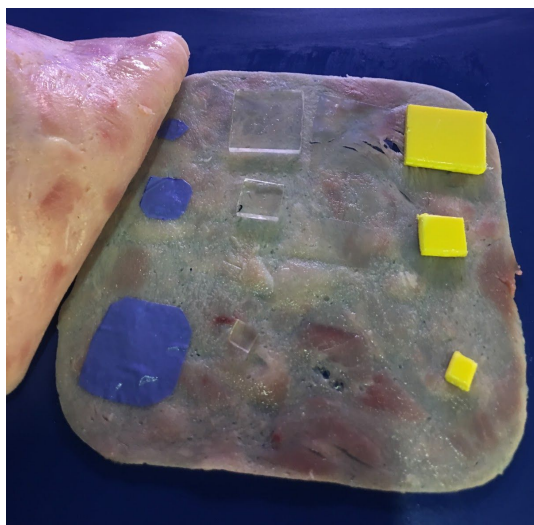


Figura 48: Muestra destapada de validación con varios cuerpos extraños

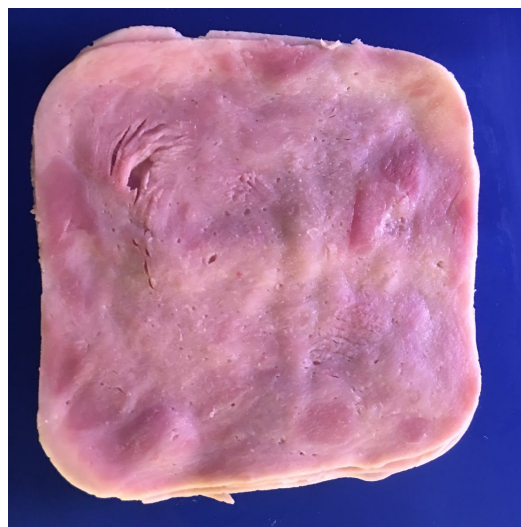


Figura 49: Sándwich de validación

Análisis de datos

Los datos obtenidos son imágenes hiperespectrales guardadas en formato ENVI, este puede ser leído por el paquete Spectral de Python. Estas imágenes son de un ancho fijo de 320 píxeles, por un alto variable normalmente unos 320 o 280 píxeles. Cada uno de estos píxeles contiene un espectro de 168 longitudes de onda. En esta ocasión los datos a usar para generar los modelos de clasificación han sido las imágenes creadas con las absorbancias. Por defecto la cámara devuelve la imagen hiperespectral RAW con la cantidad de fotones recibidos para cada longitud de onda. A diferencia de los experimentos anteriores, que usaban otro tipo de sensor, en imagen el ruido estocástico es mayor por lo que la corrección con el cálculo de la absorbancia facilita la creación de los modelos. En la siguientes imagen se muestran los diferentes espectros de cada píxel asociado.

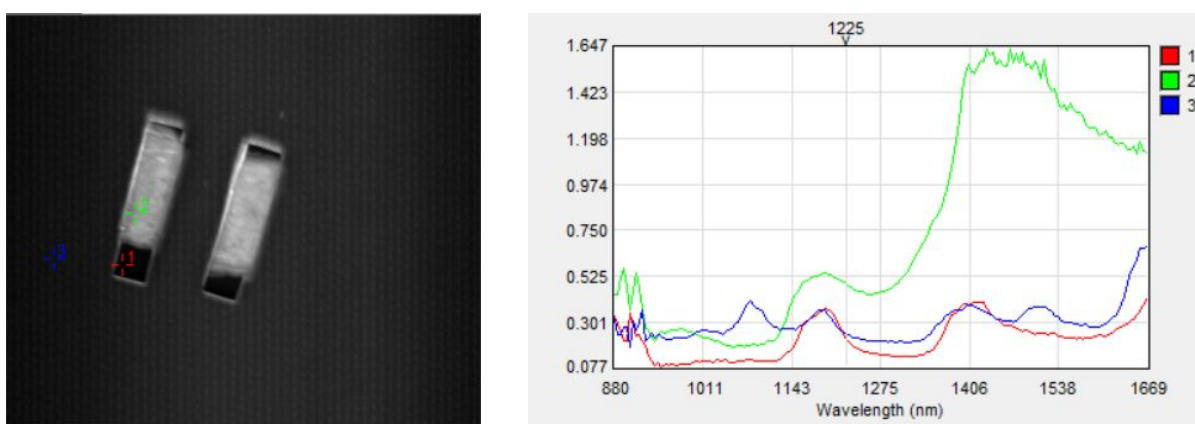


Figura 50: Imagen de sandwich jamon y PVC a la izquierda, a la derecha gráfica mostrando la absorbancia de varios píxeles en la imagen hiperespectral.

Extracción datos para modelar

Para poder entrenar las redes neuronales, es necesario obtener subconjuntos de datos suficientemente representativos. Estos tienen que representar los escenarios en que las muestras de jamón contienen cuerpos extraños de alguno de los materiales tratados. Hacer modelos con redes neuronales semejantes a las usadas con imágenes 2D RGB es aún más complejo. Se requiere de un gran número de ellas previamente clasificadas y que cubren muchos escenarios, ya sea posición tipos de muestra, cambios de iluminación etc. Puesto que esto es costoso, ya que no existe una base de datos preclasificada con imágenes hiperespectrales, el enfoque tomado es el de tratar cada píxel por separado e intentar extraer su información espectral para predecir.

Por lo tanto se pretende modelar cada píxel con los tipos de cuerpos extraños de los siguientes materiales: plástico PVC, plástico transparente PMMA, cartón, trozos de guantes plásticos y plástico tipo film transparente. Para separar según la composición, se ha usado como discriminante la distancia espectral entre picos y longitudes de onda. Fijando uno o varios umbrales para longitudes de onda se categoriza si ese espectro es de una clase u otra. Un efecto observado debido a la iluminación es el halo formado alrededor de las muestras. Este halo es una réplica sombría de la muestra, y a nivel espectral se puede interpretar como una mezcla de muestra y cinta transportadora. Este halo se ha tomado como una clase

más a predecir para evitar falsos positivos. En las siguientes imágenes se muestran coloreadas las regiones extraídas como datos de training usando la discriminación por longitud de onda. El color azul distingue los espectros relacionados con la cinta transportadora y el verde el halo formado por las sombras de la muestra.

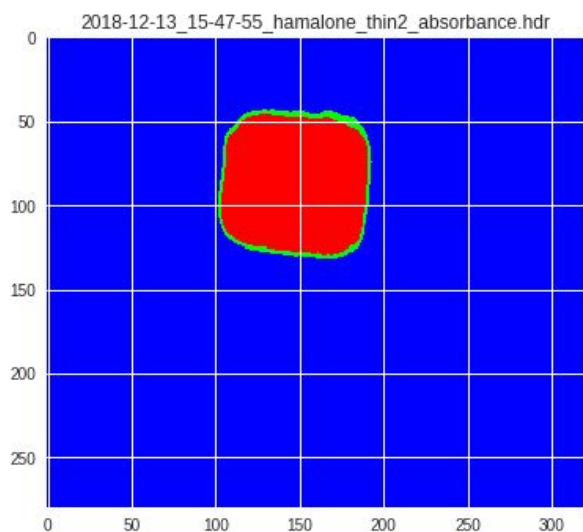


Figura 51: Rojo muestra los espectros de jamón

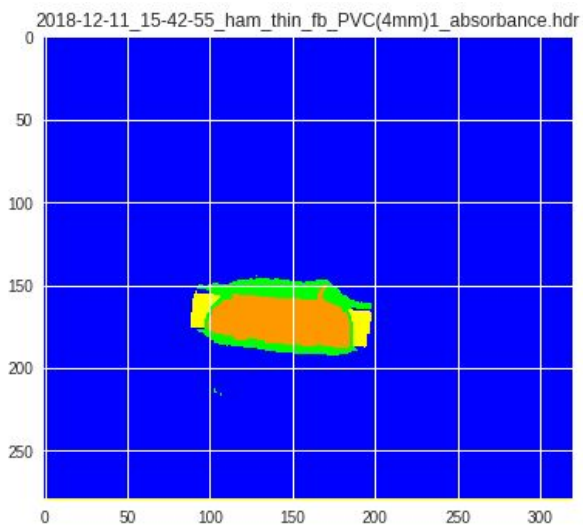


Figura 52: Amarillo plástico PVC y color anaranjado los espectros del sándwich jamón + PVC

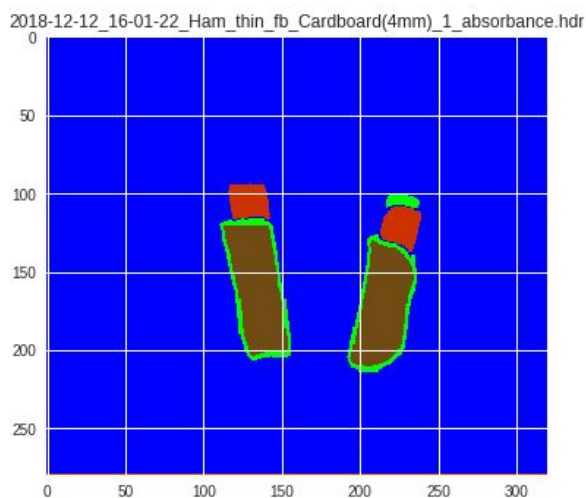


Figura 53: Marrón claro cartón, marrón oscuro sándwich jamón+cartón

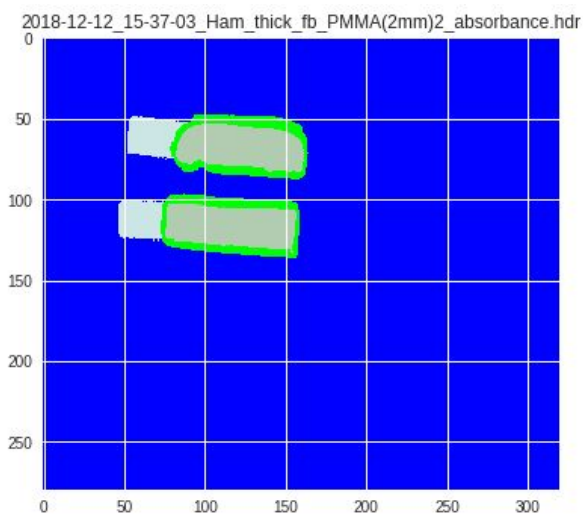


Figura 54: Gris claro PMMA, gris oscuro sándwich jamón+PMMA

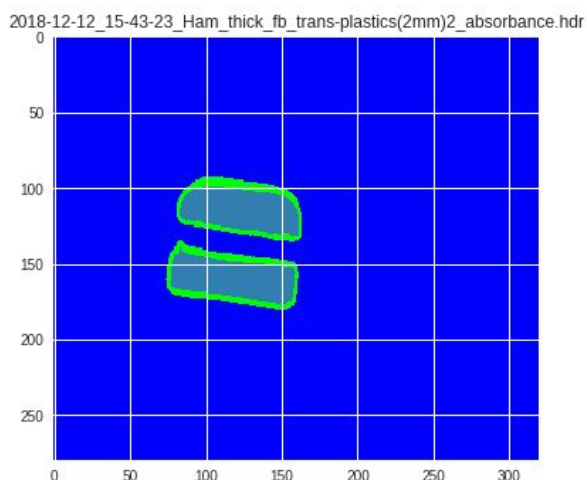


Figura 55: Azul acero para el sándwich jamón+plástico film transparente

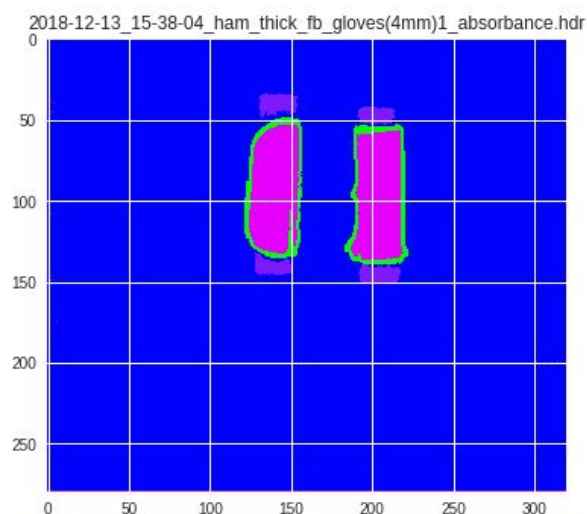


Figura 56: Violeta para los guantes plásticos y magenta para sándwich jamón+guantes

Sin contar los datos para la validación hay unos 24 GBytes en imágenes hiperespectrales. La mayor parte de las muestras para datos contienen un gran porcentaje de píxeles correspondientes a la cinta transportadora. Por lo que se han descartado la mayoría de estos intentando equilibrar el resto de clases, evitando sobreentrenar la red con datos de cinta transportadora que es lo que menos interesa. Debido a que las muestras se tomaron en varios días y algunos materiales se añadieron después, existe algo de disparidad en cuanto a volumen de espectros por tipo de muestra. En total hay 1 695 801 de espectros y el reparto final entre las distintas clases es el siguiente:

Clases	Espectros
Jamón cocido	157846
Cinta transportadora	477536
Halo	268323
Jamón + PVC	153503
PVC	43432
Jamón + Cartón	117363
Cartón	31254
Jamón + guantes plásticos	127204
Guantes plásticos	31254
Jamón + plástico film transparente	120142
Jamón + PMMA	167944

Tabla 7: Clases a modelar con el respectivo número de espectros

Clasificación usando absorbancia

Red neuronal densamente conectada

La arquitectura de la red seguida es piramidal, es decir cada capa posterior tiene un menor número de neuronas. Con esto se consigue que la extracción de características se obtenga al principio, y las capas siguientes se encarguen de generalizar hasta finalmente distinguir entre las doce posibles clases. Cuando hay tantos datos, entrenar las redes requiere mucho más tiempo y recursos. Sin embargo no es necesario un gran número de epoch para conseguir modelos predictivos aceptables, entre 5 y 10 epochs. Para tener una métrica de rendimiento del modelo antes de usarlo con los datos de validación, el conjunto de datos de training se ha separado, usando un tercio para test previo. La arquitectura mostrada consigue un 93% de exactitud con los datos de test, siendo esta la mejor de las probadas y posteriormente usada en la validación.

Layer (type)	Output Shape	Param #
=====		
dense_1 (Dense)	(None, 128)	19200

dropout_1 (Dropout)	(None, 128)	0

dense_2 (Dense)	(None, 64)	8256

dropout_2 (Dropout)	(None, 64)	0

dense_3 (Dense)	(None, 32)	2080

dense_4 (Dense)	(None, 12)	396
=====		
Total params: 29,932		
Trainable params: 29,932		
Non-trainable params: 0		

Con esta arquitectura se ha probado aplicar varios preprocesados a los datos, tal y como se hizo con las redes anteriores. Se ha probado ha aplicar varias combinaciones de suavizado, normalizar y escalado a las absorbancias, llegando a la conclusión que el mejor resultado de predicción se consigue sin aplicar ningún preprocesado. El modelo antes de usarlo con los datos de validación, se procedió a reentrenar la red esta vez incluyendo los datos que se habían apartado para test. La efectividad en la predicción del modelo se representa usando una imagen coloreada pixel a pixel según la clase predicha con la mayor probabilidad. Como hay doce clases y elegir la clase con mayor probabilidad tampoco garantiza que la decisión del modelo sea robusta, se ha incluido un umbral discriminante. Cuando la probabilidad de la clase elegida no sobrepasa el umbral, el pixel se pinta de color blanco, sino se pinta con los colores anteriormente presentados en la fase de extracción. Las tablas de a continuación muestran los resultados obtenidos en varias muestras de validación.

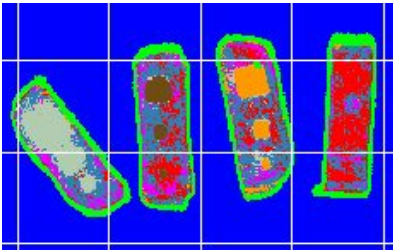
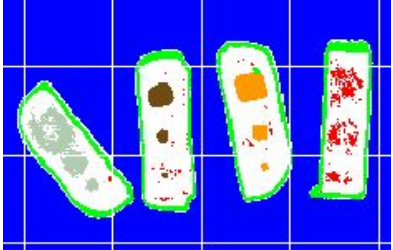

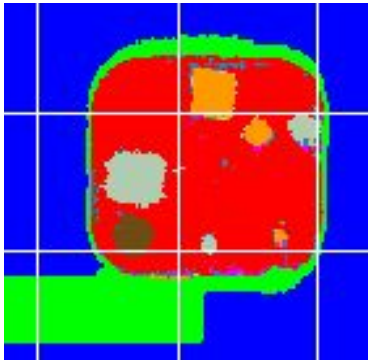
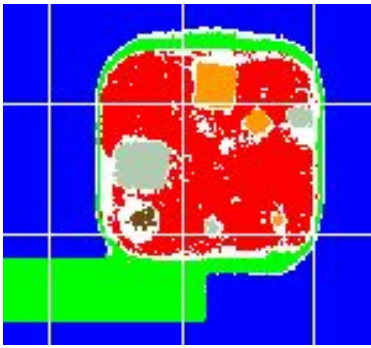

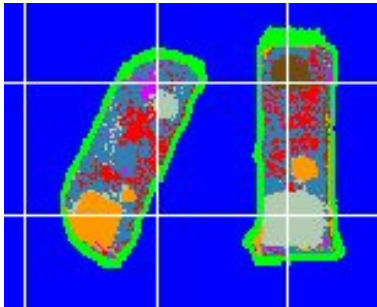
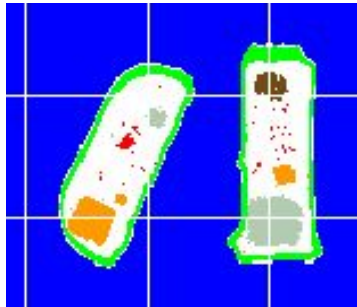
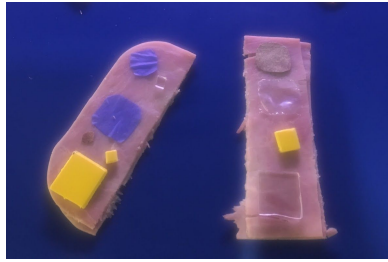
DNN - Cuerpos extraños a 2 mm de profundidad		
Umbral fijado a 0.1	Umbral fijado a 0.8	Muestras de validación al descubierto, sin las capas superiores de jamón
		
		
		

Tabla 8: Resultados del modelo con las imágenes de validación con cuerpos extraños a 2 mm de la superficie

Como se puede observar con este tipo de red neuronal se consiguen buenos resultados en la detección de cuerpos extraños que están a una profundidad de 2 milímetros sobre la superficie. Los materiales PVC, PMMA y cartón se predicen con facilidad independientemente del tamaño. Sin embargo los plásticos azules de los guantes o el plástico tipo film transparente no son detectados. Estos dos materiales son los más complejos por lo finos que son y su poca atenuación sobre el espectro principal de jamón cocido. De hecho, con el umbral a 0.1, se puede ver como los alrededores de los cuerpos extraños son confundidos con la clase de plástico tipo film transparente o la de los guantes.

El siguiente experimento, con los cuerpos extraños a 4 mm, los resultados son más interesantes. Principalmente porque los métodos analíticos usados como PLS-DA no han conseguido predecir correctamente ningún material.

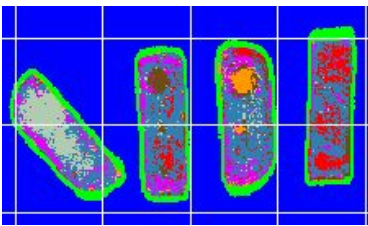
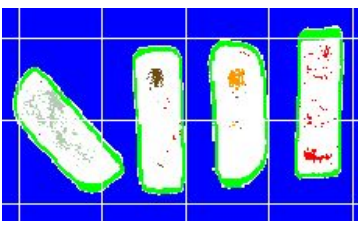

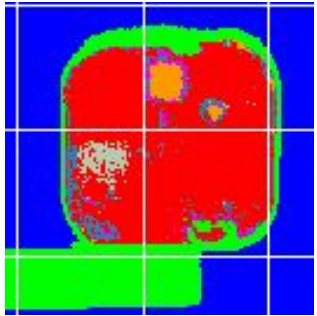
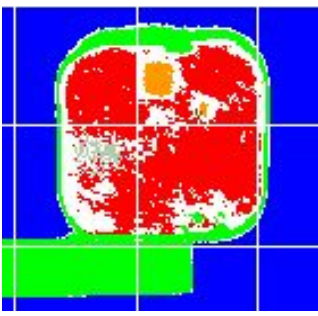

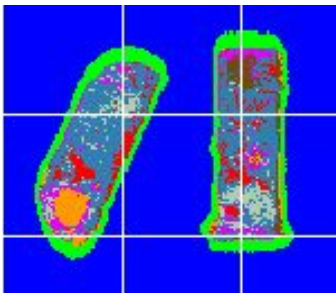
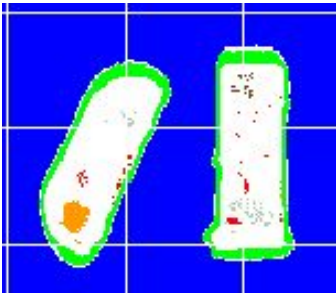
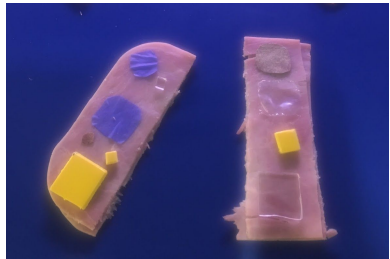
DNN - Cuerpos extraños a 4 mm de profundidad		
Umbral fijado a 0.1	Umbral fijado a 0.8	Muestras de validación al descubierto, sin las capas superiores de jamón
		
		
		

Tabla 9: Resultados del modelo con las imágenes de validación con cuerpos extraños a 4 mm de la superficie

En estas condiciones la red rinde con efectividad en los casos en que el plástico PVC es de tamaño grande y el de tamaño medio es detectado a veces. El plástico PMMA se diluye en una nube de puntos como el cartón. A pesar de ello estos puntos coinciden con la posición por lo que se puede dar como detección positiva en dos de los tres casos mostrados, en cambio tal y como pasaba en los 2 mm el resto de materiales no son detectados.

Red convolucional

La arquitectura de la red convolucional es la misma que la usada anteriormente para clasificar. Ningún tipo de preprocesado ha conseguido mejorar el rendimiento, de hecho lo empeora, tal y como pasaba en la red densamente conectada. El coste de esta red es mayor, requiere muchos más parámetros a entrenar, y tanto a la hora de entrenar como de predecir, requiere de mucho tiempo de procesado. A cambio de este sobre coste este tipo de red ha podido detectar algunos de los cuerpos extraños como los guantes o el plástico tipo film. En la siguiente tabla aparecen los resultados obtenidos con un umbral extra, el de 0.6. A diferencia con la anterior red, que con este umbral seguía sin mostrar los cuerpos extraños “difíciles”, con esta red se pueden ver nubes de puntos correspondientes a estas clases.

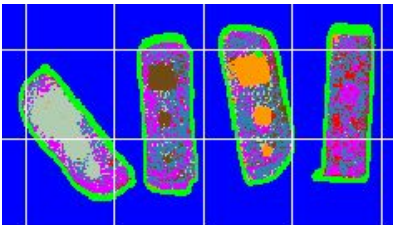
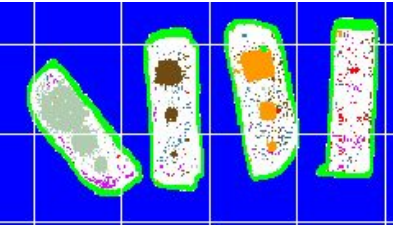
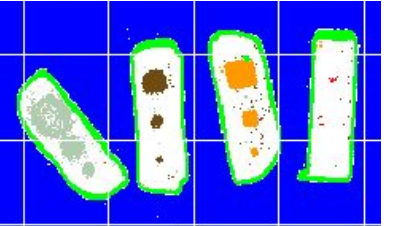
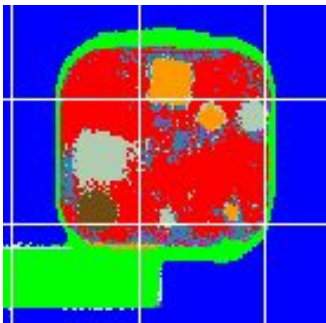
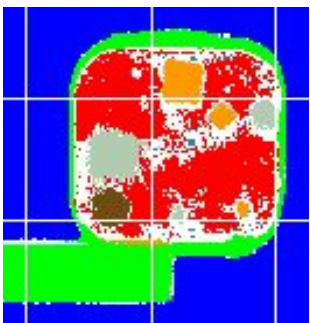
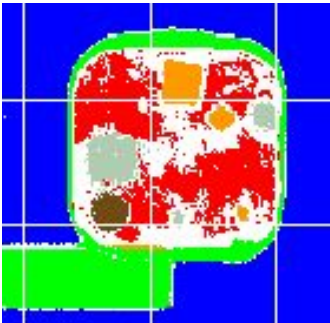
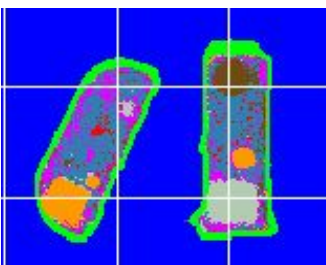
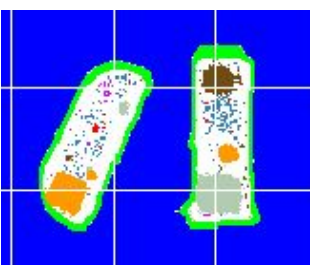
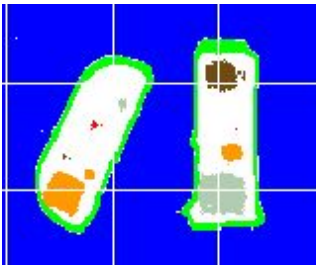
CNN - Cuerpos extraños a 2 mm de profundidad		
Umbral fijado a 0.1	Umbral fijado a 0.6	Umbral fijado a 0.8
		
		
		

Tabla 10: Resultados del modelo con las imágenes de validación con cuerpos extraños a 2 mm de la superficie

A una profundidad de 2 mm este tipo de red no ha tenido problemas en detectar y clasificar correctamente los espectros del sandwich PVC, PMMA o cartón. Con un umbral un poco más laxo, podemos ver como detecta cuerpos extraños de los tipos sándwich guantes de plástico, o plástico film. Aunque también con este umbral más bajo aparecen falsos positivos. Por ejemplo los guantes se confunden con el plástico tipo film, en términos prácticos esto tampoco es un problema ya que no importa realmente que tipo de cuerpo extraño se detecta, sino que realmente algo se detecta.


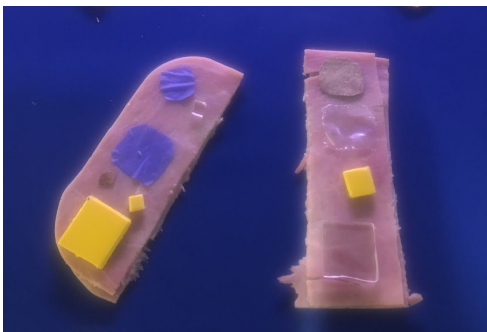
Umbral 0.6	Sándwich descubierto validación
	

Tabla 11: comparativa de predicción a 2 mm con umbral 0.6, círculo rojo mostrando la nube de puntos de cuerpo extraño tipo film transparente que en realidad no es, círculo negro nuevas predicciones correctas.

En la siguiente tabla encontramos los resultados para una profundidad de 4 mm. La predicción de que existe un cuerpo extraño de plástico PVC se vuelve más definida. Plástico PMMA mejora aunque aún no muestra siluetas definidas y el cuerpo extraño del tipo cartón aparece bastante bien definido con este tipo de red. Con el umbral a 0.6 se obtienen algunas nubes de puntos que coinciden con los cuerpos extraños de plástico tipo film transparente y el de guantes de plástico. Estos dos siguen sin estar bien definidos aunque sigue siendo un mejor resultado que con la red densamente conectada.

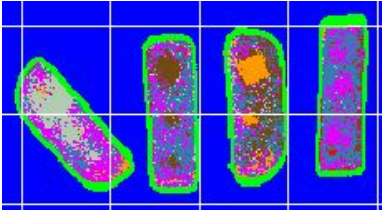
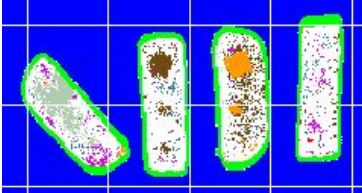
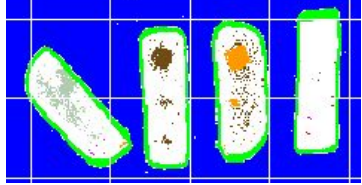
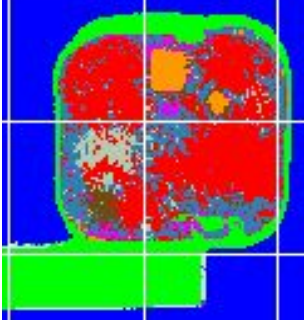
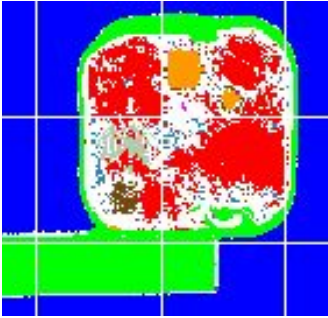

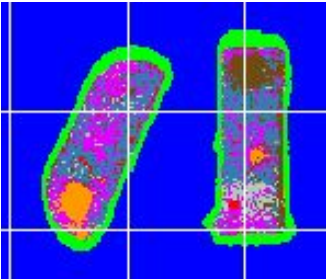
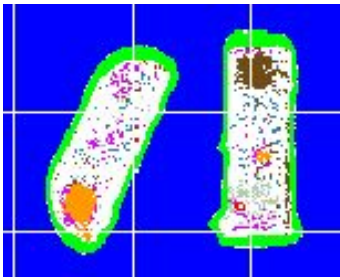
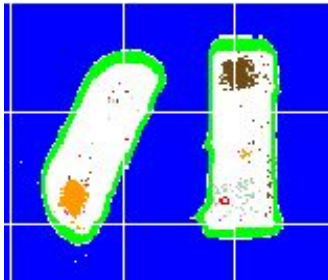
CNN - Cuerpos extraños a 4 mm de profundidad		
Umbral fijado a 0.1	Umbral fijado a 0.6	Umbral fijado a 0.8
		
		
		

Tabla 11: Resultados del modelo con las imágenes de validación con cuerpos extraños a 4 mm de la superficie

Comparativa DNN vs CNN

La siguiente tabla muestra a modo resumen los resultados obtenidos con los dos tipos de redes. Se tiene en cuenta el escenario más difícil, profundidad 4 mm, y una certeza más alta, es decir el umbral por encima de 0.8. La CNN consigue mostrar los cuerpos extraños de cartón y además define mejor los de PVC, aunque muestra un exceso de falsos positivos de cartón cuando no lo es. En cuanto a PMMA sigue siendo difícil detectarlo en ambas y el resto de tipos de cuerpos extraños ni tan siquiera aparecen. Se puede dar como parcialmente ganadora a la red convolucional, aunque hay que tener en cuenta su alto consumo en recursos. Con esta red, llegar a tener un sistema de detección de cuerpos extraños en tiempo real requerirá de un sistema más complejo y por supuesto costoso.

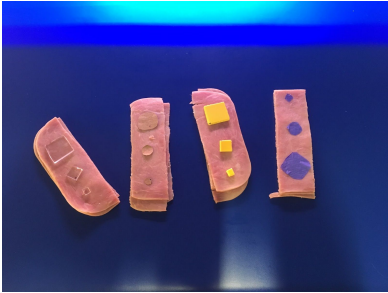
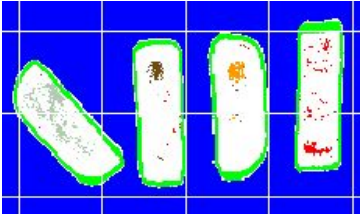
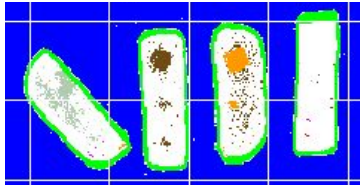



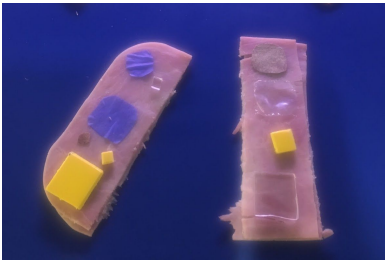
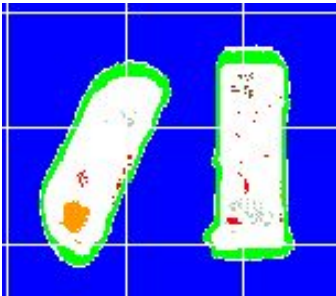
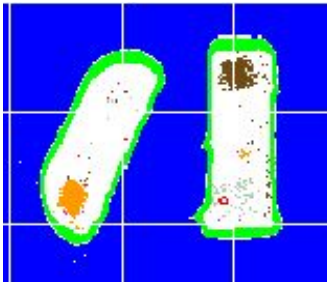
Cuerpos extraños a 4 mm de profundidad		
Validación	DNN umbral fijado a 0.8	CNN umbral fijado a 0.8
		
		
		

Tabla 12:Comparativa entre los resultados obtenidos con ambos tipos de redes en el escenario más complejo

Trabajo futuro

Las redes ya sean densamente conectadas o convolucionales han ofrecido un rendimiento superior a los métodos PLS-DA y PLS en los datos obtenidos con el Visum Palm, es decir con la espectroscopia centrada en un punto. Ahora queda demostrar que realmente estas redes funcionan mejor. Por lo que habría que conseguir nuevos conjuntos de datos para la validación de estos modelos. Si se confirma que estos modelos son más efectivos que los obtenidos con PLS o PLS-DA el paso siguiente es implantarlos. Hay que tener en cuenta que todo el análisis se ha hecho en una plataforma en el cloud, usando un hardware con bastante capacidad, por lo que implantarlo en un sistema embebido como el del Visum Palm añade un grado de complejidad. El backend de Tensorflow está ahora disponible y soportado para ser recompilado en Raspberry Pi y arquitecturas ARMv7, por lo que se abre el camino de llevar las redes neuronales a los dispositivos embebidos.

En imagen hiperespectral se han conseguido resultados no alcanzables para PLS-DA, es decir encontrar cuerpos extraños a 4 mm de profundidad. Aunque el modelo obtenido con CNN ha conseguido acercarse a detectar todos los tipos de cuerpos extraños, aún está lejos de funcionar óptimamente. Con el volumen de datos para entrenar actual, los cuerpos extraños de PVC y cartón se detectan con facilidad. En cambio habría que probar si al aumentar el número de muestras de PMMA, guantes plásticos y las de plástico tipo film transparente se consigue un modelo robusto que detecte estos cuerpos. Si este modelo predictivo se consigue, llevarlo a producción sería el siguiente paso.

Estos tests se han hecho usando la plataforma Google Colab con la arquitectura Keras->Python->Tensorflow. Hypera es un equipo de monitorización en tiempo real que tiene que capturar muchas imágenes hiperespectrales por segundo. Puesto que estas imágenes oscilan entre los 50 y 300 MBytes, hacer que estas se procesen fuera del entorno local no es viable. Es decir, enviarlas a unos servidores externos para obtener la imágenes predichas no es factible. Por el mero hecho de que en la actualidad, el ancho de banda de las telecomunicaciones es aún insuficiente. Uno de los caminos a seguir podría ser abandonar las capas de más alto nivel como Keras y Python y centrarse en la API de bajo nivel de Tensorflow en lenguaje C. Con ella se podría acceder directamente a memoria para obtener las imágenes desde los controladores de las cámaras, y usar el backend de Tensorflow para realizar los cálculos requeridos en las GPU necesarias.

Conclusiones

El objetivo principal de esta tesis era comprobar si las redes neuronales funcionan tan bien o mejor que los métodos de análisis clásicos de la quimiometría como Partial Least Squares o Partial Least Squares Discriminant Analysis. En cada uno de los experimentos mostrados las redes neuronales han sido superiores a estos métodos. Usando el conjunto de datos de espectros focalizados para clasificar se consiguen porcentajes de exactitud del 99.93% con los datos de testeo. Con un modelo PLS-DA se puede llegar a un 44% de exactitud y si se quiere mejorar este comportamiento es necesario hacer varios con menos clases y realizar una predicción en cascada.

Otro de los objetivos que también se ha cumplido, ha sido el de ver el rendimiento de los diferentes tipos de redes con muy pocos datos para entrenar. Tanto para clasificar como para cuantificar el volumen de espectros era bajo, menor a 1000 espectros. Sin embargo, aumentando el número de veces que estos datos pasan por las redes, las epochs, se consiguen modelos con alto grado de exactitud. Con los conjuntos de datos para cuantificar además de cumplir con el objetivo de modelar con pocos datos también se comprueba cómo de efectivas son las redes usando datos raw. Para este tipo de sensores, que tienen menos ruido estocástico, utilizar los espectros raw y no las absorbancias, conjunto con las redes convolucionales parece ser una opción viable. Con los datos raw aplicando una serie de preprocesados típicos se consiguen las métricas de predicción más favorables, un RMSE con los datos de test de 0.96 en contraposición del 1.91 obtenido con un modelo PLS usando absorbancias.

El poder usar los datos raw directamente en equipos handheld supone un ahorro de tiempo considerable. En la actualidad hay que recalibrar cada pocas muestras el equipo con un spectralon y así poder recalcular las absorbancias correctamente. Haber cumplido con este objetivo con la imagen hiperespectral también hubiese sido un gran avance, por desgracia los mejores resultados obtenidos con redes neuronales y datos raw, con o sin preprocesados, distan muchísimo de los obtenidos con absorbancias. Por lo que por el momento con la tecnología actual es más conveniente continuar usando la calibración del equipo, por suerte para estos la calibración puede durar varias horas sin renovarse.

Donde realmente se ha sacado y se conseguirá sacar más partido de las redes neuronales y la plataforma usada en general, es con la imagen hiperespectral. El volumen de datos y los tiempos de respuesta tan bajos que se requieren con este tipo de aplicaciones, sí que justifica la necesidad de usar redes neuronales. Implantar en el sistema una plataforma como Tensorflow que abstraer todos los entresijos de los cálculos y el control de GPUs para obtener los resultados en menor tiempo es primordial. En la actualidad IRIS Technology Solutions cuenta con herramientas de análisis PLS-DA que pueden ser ejecutadas en GPU. Estas herramientas no son simplemente difíciles de escalar y sobretodo de mantener, sino que como se ha mostrado en los ejemplos, difícilmente llegarán a ofrecer unos resultados tan satisfactorios como con la plataforma presentada.

En definitiva, para llegar a los requisitos que se proyectan en las PAT y en la industria 4.0 es necesario usar nuevas tecnologías analíticas. Puesto que la plataforma presentada ha obtenido buenos resultados en cada uno de los experimentos el paso restante es el de su implantación.

Referencias

- [1] Gorecky, Dominic & Schmitt, Mathias & Loskyll, Matthias & Zühlke, Detlef. (2014). Human-machine-interaction in the industry 4.0 era. 289-294. 10.1109/INDIN.2014.6945523.
- [2] Bakeev, K.A. (2010) Process Analytical Technology: Spectroscopic Tools and Implementation Strategies for the Chemical and Pharmaceutical Industries. John Wiley & Sons ISBN: 0470689609, 9780470689608
- [3] B Marić, Ljiljana & Jovic, Branislav & D Petrović, Slobodan & M Nikolić, Aleksandar & J Homšek, Irena. (2014). The application of NIR spectroscopy with chemometric analysis for monitoring a powder blending process. Journal of the Serbian Chemical Society. 79. 10.2298/JSC010413080M.
- [4] K.-J. Baik, J. H. Lee, Y. Kim, B.-J. Jang, Pharmaceutical tablet classification using a portable spectrometer with combinations of visible and near-infrared spectra, in: Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on, IEEE, 2017, pp. 1011– 1014.
- [5] P. Geladi, B. R. Kowalski, Partial least-squares regression: a tutorial, *Analytica chimica acta* 185 (1986) 1–17.
- [6] Liu, Jinchao et al. Deep Convolutional Neural Networks for Raman Spectrum Recognition: A Unified Solution. CoRR abs/1708.09022 (2017): n. pag.
- [7] Bjerrum, Esben Jannik et al. Data Augmentation of Spectral Data for Convolutional Neural Network (CNN) Based Deep Chemometrics.” CoRR abs/1710.01927 (2017): n. pag.
- [8] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Preenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: experiences from the scikit-learn project,” in ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108–122, 2013.
- [9] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. "TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc."
- [10] F. Chollet et al., “Keras.” <https://keras.io>, 2015.
- [11] EliteDataScience. 2017. Keras Tutorial: The Ultimate Beginner’s Guide to Deep Learning in Python. [ONLINE] Disponible en: <https://elitedatascience.com/keras-tutorial-deep-learning-in-python>. [Consulta 16 Octubre 2018].

- [12] Torres,J, (2018), Deep learning Introducción práctica con Keras, Barcelona, Creative Commons Disponible en: <https://torres.ai/redes-neuronales-densamente-conectadas/>
- [13] Wikipedia (2018). Perceptron [ONLINE]. Disponible en <https://en.wikipedia.org/wiki/Perceptron> [Consulta 22/12/2018]
- [14] Karpathy. 2018. Convolutional Neural Networks for Visual Recognition. [ONLINE] Disponible en: <http://cs231n.github.io/neural-networks-1/>. [Consulta 16 Octubre 2018].
- [15] Jovian Lin. 2018. categorical_crossentropy VS. sparse_categorical_crossentropy. [ONLINE] Disponible en: <https://jovianlin.io/cat-crossentropy-vs-sparse-cat-crossentropy/>. [Consulta 25 Octubre 2018].
- [16] Nils Ackermann. 2018. Introduction to 1D Convolutional Neural Networks in Keras for Time Sequences. [ONLINE] Disponible en: <https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf>. [Consulta 1 Noviembre 2018]
- [17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, pp. 1929– 1958, Jan. 2014.
- [18]Yahwes. 2016. MAE and RMSE — Which Metric is Better?. [ONLINE] Disponible en: <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>. [Consulta 20 Noviembre 2018].
- [19] A. Savitzky, M. J. Golay, Smoothing and differentiation of data by simplified least squares procedures., Analytical chemistry 36 (8) (1964) 1627–1639.

Glosario

Absorbancia: En espectrofotometría, la absorbancia se define como $Abs(\lambda) = -\log_{10}(I/I_0)$ donde I es la intensidad de la luz con una longitud de onda específica tras haber atravesado una muestra, I_0 es intensidad de la luz antes de entrar a la muestra (intensidad de la luz incidente)

Batch size: es el número de ejemplos del conjunto de datos de entreno que pasan hacia delante y hacia atrás por la red. Cuanto mayor es este número más memoria se necesita.

Bias: indica cómo de lejos están nuestras predicciones de los valores reales. Estas neuronas se pueden ordenar en diversas topologías.

Bright: espectro tomado con una fuente de luz que irradia un material de reflectancia conocida, como un spectralon. Sirve para obtener el rango superior esperable que un sensor espectroscópico puede ofrecer.

CNN: siglas del inglés Convolutional Neural Network, las redes neuronales convolucionales consisten en múltiples capas de filtros convolucionales de una o más dimensiones. Después de cada capa, por lo general se añade una función para realizar un mapeo causal no-lineal.

Dark: espectro tomado sin fuente de luz con un equipo espectroscópico para saber el ruido estocástico del sensor.

Deep learning: es un conjunto de algoritmos de clase aprendizaje automático que intenta modelar abstracciones de alto nivel en datos usando arquitecturas compuestas de transformaciones no lineales múltiples.

Dense layer: capa de neuronas en una red neuronal. Cada neurona recibe un valor de entrada de todas las neuronas de la capa anterior por lo tanto está densamente conecta.

Epoch: cuando todos los datos que se usan para entrenar el modelo han pasado por la red neuronal.

GPU: una unidad de procesamiento gráfico o GPU (*graphics processing unit*) es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central.

Keras: es una biblioteca de red neuronal de código abierto escrita en Python. Es capaz de ejecutarse sobre TensorFlow, Microsoft Cognitive Toolkit o Theano.

K-fold Cross-Validation : los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto ($K-1$) como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba.

Layer: capa de neuronas en una red neuronal, una capa tiene una matriz de pesos \mathbf{W} , un vector de Bias \mathbf{b} y las activaciones de las capas previas \mathbf{a} .

Leave-One-Out Cross-Validation: implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento.

Loss: valor escalar a minimizar mientras se entrena el modelo, cuanto más pequeño es más cerca está el modelo de las predicciones correctas

MAE: siglas del inglés Mean Absolute Error, es una medida de la diferencia entre dos variables continuas.

NIRS: siglas del inglés Near-Infrared Spectroscopy, es un método espectroscópico que usa la región cercana al infrarrojo del espectro electromagnético (de 780nm a 2500nm), tiene aplicaciones en áreas como la farmacéutica, la alimentación, control de calidad agroquímico, química atmosférica entre otros

Outlier: Un valor más extremo es un valor en un conjunto de datos que es muy diferente de los otros valores. Esto es, los outliers son valores excepcionalmente lejanos del centro.

PLS-DA: siglas del inglés Partial Least Squares Discriminant Analysis es una variante de PLS que se utiliza cuando la Y es binaria.

PLS: siglas del inglés Partial Least Squares, es un método estadístico que encuentra una regresión lineal mediante la proyección de las variables de predicción y las variables observables a un nuevo espacio.

Python: lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Raw: nombre que recibe un espectro tomado de una muestra, este contiene el número de fotones recibidos por cada canal en un tiempo de muestreo determinado.

ReLU: siglas del inglés Rectified Linear Unit en el contexto de las redes neuronales artificiales, el rectificador es una función de activación definida como: $f(x)=\max(0,x)$

RMSE: siglas del inglés Root Mean Squared Error, es una medida frecuente de la diferencia entre valores predichos por un modelo o por un estimador y los valores observados.

RMSProp: versión mejorada del optimizador SGD

SGD: siglas del inglés Stochastic Gradient Descent, es un tipo de optimizador usado para encontrar un mínimo global en la función de pérdida tiene el valor mínimo posible.

Spectralon: es un fluoropolímero el cual tiene la más alta reflectancia difusa de cualquier material conocido sobre las regiones del espectro ultravioleta, visible y del infrarrojo cercano.

Tensorflow: es una biblioteca de código abierto para aprendizaje automático desarrollado por Google capaz de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos.

Tensors: podemos decir que un tensor, en TensorFlow, describe una matriz numérica multidimensional, con una colección de datos de cero o n dimensiones, determinada por rango, forma y tipo.

Test: conjunto de datos que se separan del global y se usan para comprobar la robustez de un modelo predictivo.

Training: conjunto de datos que se separan del global y se usan para generar un modelo predictivo.

Weights: muestran la fortaleza con la conexión entre neuronas, es decir si se incrementa la entrada cuánto influye en la salidas.